

Szegedi Innovatív Informatika Verseny

2023



SZIIV Program (2023. április 21. – péntek)

Időpont	Műszaki informatika szekció	Informatika szekció
8:30–	Regisztráció	
9:00–9:15	Megnyitó	
9:20–11:00	Előadások I.	
9:20–9:40	EcoLocator: EcoLocator	SWP: Családfa-számítás
9:40–10:00	Team Venom: Projekt Venom	Asszem...bili: A világ első Brainfuck operációs rendszere (FirstOS)
10:00–10:20	Kandó Racing Team: KRT Telemetry Systems	TeachHelper: Arcfelismerő alkalmazás fejlesztése jelenlét ellenőrző rendszerhez
10:20–10:40	Fizikus módszer: Optimális billentyűkiosztás tervezése magyar nyelvhez	vajdad4m: KeySniff
10:40–11:00	SolarWise: Hordozható szigetüzemű napelemes MPPT rendszer kifejlesztése	NeuMapp: NeuMapp
11:00–11:20	SZÜNET	
11:20–13:00	Előadások II.	
11:20–11:40	Robbineers: Tisztább iskolát!	Fizikus módszer: MateKIT – matematikai oktatóprogram
11:40–12:00	TGAM: TGAM projekt	RepairSync: RepairSync
12:00–12:20	vajdad4m: Testkulcspontok becslése FMCW radarképek segítségével	Rostás Raul: Biztonságos és privát digitális kommunikáció, decentralizált hálózatokon keresztül
12:20–12:40	Péter Attila: Arduino vezérelt automatikus öntözőrendszer	TalkHub: TalkHub – Csapatmunka távolról
12:40–13:00		chmod 777: LEX
13:00–14:00	Ebédszünet	
13:30–14:00	Projektbemutató előkészületek	
14:00–15:00	Projektbemutatók	Projektbemutatók
16:00–17:00	Eredményhirdetés	

Tartalomjegyzék

<i>SZIIV Program</i>	1
<i>Előszó</i>	5
<i>Programbizottság</i>	6
<i>Műszaki informatika szekció</i>	9
Kónya Richárd:	
EcoLocator	11
Barkóczy Bálint:	
Projekt Venom	15
Kovács Tamás, Váradi Marcell, Kiss Csaba:	
Telemetry Systems	19
Kovács János:	
Optimális billentyűkiosztás tervezése a magyar nyelvhez	23
Varga Bence, Jordán Nikolasz:	
Hordozható szigetüzemű napelemes MPPT rendszer kifejlesztése	27
Bulyák Balázs, Kertész Levente, Soós Tamás:	
Tisztább iskolát!	31
Vadász Balázs:	
TGAM projekt	35
Vajda Ádám:	
Testkulcspontok becslése FMCW radarképek segítségével	39
Péter Attila:	
Arduino-vezérelt automatikus öntözőrendszer	43
<i>Informatika szekció</i>	47
Török Péter, Kocsis Nándor, Domokos Ádám:	
Családfa-számítás	49

Bábel Edvárd István:	
A világ legelső Brainfuck operációs rendszere (FirstOS)	51
Csizmadia Ádám, Szalai Áron:	
Arcfelismerő alkalmazás fejlesztése jelenlét ellenőrző rendszerhez	55
Vajda Ádám:	
KeySniff: Gépelt szöveg feltárása billentyűzet hangfelvételekből mély tanulásal	59
Hazafi Gergő:	
NeuMapp	63
Kovács János:	
MateKIT – matematikai oktatóprogram	67
Magyar Virág, Laczkó Örs:	
RepairSync	71
Rostás Raul:	
Biztonságos és privát digitális kommunikáció, decentralizált hálózatokon keresztül	73
Szabó Marcell:	
TalkHub – csapatmunka távolról	77
Márton Gábor:	
LEX	81
<i>Együttműködő partnereink</i>	<i>85</i>

Előszó

A Szegedi Innovatív Informatika Verseny 2023-ban kilencedik alkalommal kerül megrendezésre, ahol a középiskolás diákok mutathatják be megvalósított innovatív ötleteiket. A versenyt a Szegedi Tudományegyetem Természettudományi és Informatikai Karán belül az Informatikai Intézet indította útjára.

A verseny keretein belül ezúttal sem kötöttük meg a diákok kezét és kreativitását, bármilyen, őket foglalkoztató informatikai témájú pályamunkával nevezhettek az alábbi két szekció valamelyikére:

A **Műszaki informatika** szekcióba olyan pályamunkák beérkezését vártuk, melyeknél csupán egyetlen kikötésünk volt: a feladat ne csak programozási feladatból álljon. A diákoknak meg kellett építeni, vagy már meglévő elemekből össze kellett állítani egy rendszert, amely működtetéséhez szükséges szoftvert is a nekik kellett elkészíteni.

Az **Informatika** szekcióba olyan pályamunkákat vártunk, amelyekben a diákok egy elkészített szoftvert mutatnak be. A fejlesztéshez tetszőleges programozási nyelv és programozói függvénykönyvtár használható volt. A szoftver kategóriáját illetően nem tettünk megkötést, lehetett játék, asztali számítógépen futtatható alkalmazás, webes alkalmazás, mobil applikáció, vagy Kinect-es alkalmazás.

Az idei versenyre összesen 39 pályamunka érkezett be. Versenyünk kétfordulós, az első fordulóra beérkezett pályamunkából a zsűri szekciónként a legjobb 10-10 pályamunkát benyújtó csapatoknak ad lehetőséget, hogy a döntőben bemutassák elért eredményeiket. Remélhetőleg idén ismét előben találkozhatunk a diákokkal és ismerhetjük meg alkotásaikat.

A rendezvényt idén is számos neves informatikai és műszaki cég támogatta, amelyért ezúton is hálásak vagyunk. A zsűri által legjobbnak ítélt pályamunkákat értékes és hasznos díjakkal jutalmazzuk. A verseny szervezői számára kiemelten fontos, hogy ne csupán a csapatok tagjait és felkészítő tanáraikat, de iskoláit is díjazzuk, hogy ily módon támogassuk tehetséggondozó tevékenységüket.

Szeged, 2023. április

Dr. Németh Gábor
a SZIIV verseny szervezője

Programbizottság

Műszaki informatika szekció

A zsűri elnöke:



Dr. Pletl Szilveszter
főiskolai tanár
SZTE Műszaki Informatika Tanszék

A zsűri tagjai:



Dr. Mingesz Róbert
adjunktus
SZTE Műszaki Informatika Tanszék



Dr. Kincses Zoltán
adjunktus
SZTE Műszaki Informatika Tanszék



Vékony László
senior üzleti elemző
Sonrisa Technologies



Trauer János
középiskolai tanár (fizika-informatika szak)
Makói József Attila Gimnázium, Makó

Informatika szekció

A zsűri elnöke:



Dr. Nyúl László
intézetvezető, tanszékvezető egyetemi docens
SZTE Képfeldolgozás és Számítógépes Grafika Tanszék

A zsűri tagjai:



Dr. Vinkó Tamás
egyetemi docens
SZTE Számítógépes Optimalizálás Tanszék



Dr. Pflanzner Tamás
adjunktus
SZTE Szoftverfejlesztés Tanszék



Téglásy Nóra
szoftverfejlesztő
ExxonMobil Hungary Kft.



Fodor Zsolt
középiskolai tanár (matematika-fizika-informatika szak)
Baranya Megyei SZC Komlói Technikum, Komló

Szervezők



Dr. Nagy Antal
egyetemi docens, intézetvezető helyettes
SZTE Képfeldolgozás és Számítógépes Grafika Tanszék



Dr. Németh Gábor
adjunktus
SZTE Képfeldolgozás és Számítógépes Grafika Tanszék



Dr. Kincses Zoltán
adjunktus
SZTE Műszaki Informatika Tanszék



Tóth-Ságiné Kun Eszter
közkapcsolati szakértő
SZTE Informatikai Intézet

Műszaki informatika szekció

EcoLocator

EcoLocator

Kónya Richárd

Felkészítő tanár: Bicskei Róbert

SZSZC Déri Miksa Műszaki Technikum, 6724 Szeged, Kálvária tér 7.

1. Bevezetés

Az elmúlt évek fejlődése az informatikai és technológiai területen lehetővé tette, hogy olyan eszközöket fejlesszünk ki, amelyek kisegítenek bennünket a mindennapi teendőinkben, valamint leveszik a legtöbb triviális vagy fárasztó cselekvés terheit a vállunkról. Mire haza érünk egy robotporszívó már fel is takarított, a vacsorát, amit a mobiltelefonunkon rendeltünk legföljebb kettő gombnyomással, pillanatokon belül ki is hozzák, a másnapi beszámolónkat pedig egy mesterséges intelligencia másodpercek alatt legenerálja és még tanácsokat is ad az előadásunkhoz. Hogy hogyan jutunk el a város egyik feléből a másikba csak a bőségzavar miatt okoz fejtörést, azonban ez nem mindenki számára ennyire kézenfekvő. Rengeteg embernek látásprobléma miatt kihívást jelentenek olyan hétköznapi feladatok, mint leszaladni a kisboltba, vagy eljutni a buszmegállóba. Ez a gondolatmenet alapján jutottam arra következtetésre, hogy a látássérültek számára szeretnék kifejleszteni egy segítő eszközt.

2. A probléma megoldása

2.1. A cél

Mielőtt belekezdtem volna a projektembe, azon töprengtem, hogyan lehetne hasznosítani a technológiát, ami körülvesz minket a mindennapokban, hogy segítsünk azokon, akiknek nagy szüksége lenne a támogatásra. Ennek kapcsán jött szóba az EcoLocator, egy kézi eszköz, amely hangjelzésekkel tájékoztatja a látássérülteket a lábuk előtt lévő akadályokról, így egy modern alternatívát kínálva a megszokott fehér bot helyett.

2.2. Az EcoLocator működése

Az EcoLocator egy zseblámpa méretű kézi eszköz, amelyet egy Arduino mikrokontroller vezérel. Egy ultrahangos érzékelő segítségével meghatározza az előtte lévő felület távolságát legfeljebb 3 méterig, és a közelség függvényében egyre gyorsabban csipog, rezeg, valamint jelzőfényt bocsájt ki, valamint jelezi egy akadály felbukkanását. A hangeffektusokat egy fülhallgatón keresztül hallja a felhasználó, amit egy jack aljzaton keresztül csatlakoztat. Az eszköz egy powerbank-kal, USB-C csatlakozón keresztül használható.

2.3. A tervezés

Az ötlet kitalálása után a megfogalmazott igények kielégítésére szolgáló alkatrészek meghatározása volt. Mikrovezérlőnek egy Arduino Nano-t választottam, mivel ár-érték arányban az volt a legkedvezőbb, valamint a méretéből és kialakításából adódóan könnyű beleépíteni az áramkörbe.

Távolságmérésre egy HC-SR04-nek keresztelt ultrahangos szenzort találtam, ennek előnye, hogy bármilyen felületről megbízhatóan megállapítja annak távolságát. A szenzor 2mm-en belül és 3000mm-en kívül nem mér, hanem egy alaphálót beállított számot ad vissza, de ezt az Arduino programjában könnyen kezelni lehet.

Hangrögzítésre egy ISD1730py chipet használtam, ennek számos előnye is van, például, hogy több, egymástól független szöveget fel lehet rá venni, valamint azokat kedv szerint visszajátszani. Emellett két különböző alkalmazási módja van, az egyik az önálló, a másik SPI, vagy vezérelt működtetés. Az önálló működés lehetőséget ad arra, hogy a felhasználó nyomógombokkal vezérelje a chipet, például a felvételeket így készítettem el egy mikrofon segítségével. A vezérelt módot már beszerelt állapotban érdemes alkalmazni, ugyanis az Arduino-nak programjába be lehet emelni egy SPI könyvtárat, ami kifejezetten ezekkel a parancsokkal foglalkozik. A chip kimenetét pedig 3,5mm-es jack aljzatra vezetem ki, amibe a fülhallgatót bedugva hallani lehet a felvett szöveget.

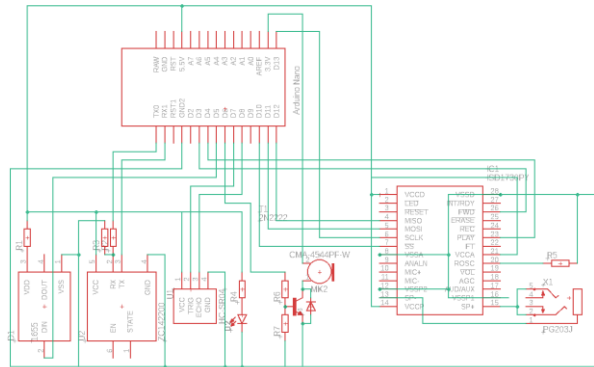
Jelzőfénynek egy WS2812B típusú LED-szalagot választottam IP65-ös érintésvédelemmel. Ez a LED-szalag azért különleges, mert minden egyes fényforrást külön lehet vezérelni az Arduino NeoPixel nevű könyvtárának segítségével, így elérhető egyértelműen megjeleníthető rajta a mért távolság. Az érintésvédelem azért volt fontos szempont, mivel az eszköz használatából adódóan ki van téve a felhasználó kezének, valamint az azon lévő izzadságnak.

Hogy tovább bővítssem az érzékszervi visszajelzéseket, egy rezgőmotort is beépítettem, ami párhuzamosan működik a csipogással. Ezzel az alkatrésszel csak annyi probléma volt, hogy ez legfőljebb 3,5V-ról működik, ameddig a többi elem 5V-ról, emellett jelentősen nagyobb az áramfelvétele, mint amit az Arduino digitális lába tud biztosítani. Mindezt a motor működését a következőképpen oldottam meg. A motor anódját az Arduino 3,3V-os lábára, a katódját pedig egy BD139-es tranzisztor kollektorára kötöttem, a tranzisztor bázisát pedig egy PWM modulációra képes digitális lábra, ezzel már vezérelni is lehet a rezgést.

Az eszköz ezek mellett tartalmaz még egy HC-06-os Bluetooth modult is, amelyen keresztül a mért adatokat lehet lekérdezni. Ehhez egy telefonos alkalmazást is készítettem.

Miután az alábbi alkatrészeket összeszedtem, megterveztem az 1. ábrán látható kapcsolási rajzot, hogy szépen, egyben lássam át az egész projektet. Ezt követően elkezdtem a beszerzést, volt, amit külföldről rendeltem, az alapvető áramkörti elemeket, mint az ellenállásokat helyi

kiskereskedésben találtam meg, de például a rezgőmotort egy régi, már nem működő telefonomból szereltem ki.



1. ábra Kapcsolási rajz

2.4. A program

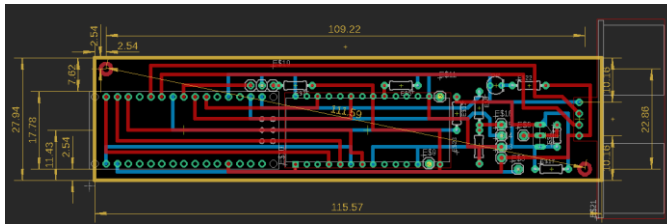
A program megírását a működési elvének meghatározásával kezdtem. Először az ultrahangos távolságmérőből érkező adatokat kell átváltani, ugyanis az a visszaverődő hang által megtett út idejét adja meg milliszekundumban, ezt kell megszorozni a hang levegőben való terjedési idejének felével, mivel csak az oda utat számoljuk. Ezt az (1)-es képletből állítottam elő.

$$távolság = 0,01723 * t \quad (1)$$

A következő kérdést az akadályok felmérése hozta. Alapnak mindig az utolsó harminc mérést vettem, ebből az utolsó 10 átlagát viszonyítottam az előtte levő 20 átlagához. Ha hirtelen nagy mértékű eltérést érzékel az eszköz, akkor az eszköz kimondja az „Akadály” szót. A LED-ek 50cm-enként villannak fel, az első, a fehér pedig csak akkor, ha méréshatáron kívül mér az eszköz.

2.5. Fizikai megvalósítás

Az első tesztelések után megterveztem a nyomtatott áramkört, ami először egy zseblámpa testébe került bele a teszteléshez, majd a közeljövőben az iskolában 3D nyomtatóval gyártattatok neki egy testhezálló vázat. Az áramkör a 2. ábrán, az összeszerelt eszköz pedig a 3. ábrán látható.



2. ábra Nyomtatott áramkör



3. ábra Az összeszerelt eszköz

2.6. További fejlesztések

Az eszköz kialakítása számos további funkció hozzáadására ad lehetőséget, például a Bluetooth kapcsolattal az eszköz összes számítási feladatát ki lehetne váltani egy mobiltelefonnal.

3. Elért eredmények

Az EcoLocator jelentős lépés lehet az esélyegyenlőség teremtés világában, ugyanis egy létező problémára kínál modern megoldást, megfizethető áron. Tervezése során rengeteg új dologgal szembe jöttem, ötlettel gazdagodtam, valamint sok mindent megtanultam. Az eszköz jelenlegi verziója rengeteg változaton keresztül ment, mindig egyel több funkció, másik alkatrész érződött szükségesnek a projekt teljességéhez, azonban jelenlegi állapotában merem állítani, hogy megállja a helyét, erről tanúskodik a XXVI. Országos Elektronikai Konstruktív Versenyen elért eredményem az EcoLocatorral.

Projekt Venom

Team Venom

Barkóczy Bálint

Felkészítő tanár: Bicskei Róbert

SZSZC Déri Miksa Műszaki Technikum, 6724 Szeged, Kálvária tér 7.

1. Bevezetés

Mai világunkban a radioaktivitás és a hasadó anyagok mindennapossá váltak, mégis rengeteg tévhit és misztérium övezi a témát. Aki érdekelt a dologban, az is nehezen juthat műszerekhez és nincs biztosítva minden eszköz számára. Ezen okok miatt saját felszerelés elkészítésével kezdtem kísérletezni, célom lett, hogy ez a világ az átlagember számára is elérhető legyen. Az egyik nélkülözhetetlen mérőműszer a gamma spektrométer, mely az izotóp által kibocsátott sugarak energiája alapján megállapítja a forrás típusát anyagi szinten. Ezt úgy teszi, hogy egy szcintillátor jeleit vizsgálja, melyek az energia nagyságát képviselik fényvillanások formájában. A Projekt Venom egy ilyen műszer, ami kompakt módon magába foglalja a detektort (szcintillátort) (2. ábra), a feldolgozáshoz szükséges áramkört és mikrovezérlőt. Ezzel bármilyen izotópot beazonosíthatunk.

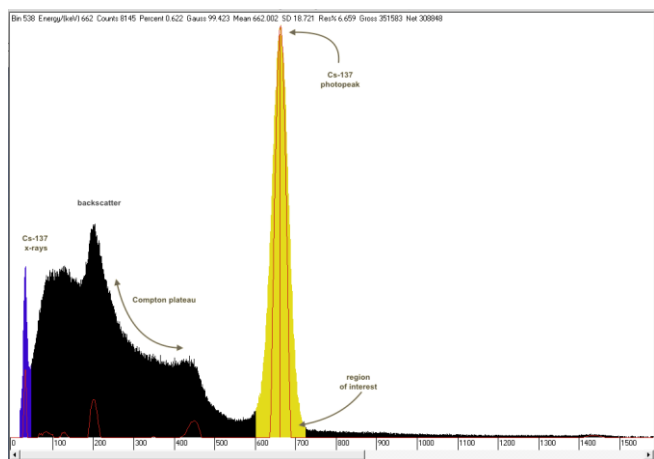
2. A probléma megoldásának menete

A szcintillátorból kilépő fotonok gyengék és kevesen vannak, így elektromosan kell őket erősíteni egy PMT csővel. A PMT pulzusait szintén erősíteni kell, lehetőleg torzításmentesen. A végleges analóg jelet egy vezérlő egység digitalizálja, elemezi és tárolja későbbi megjelenítés céljából. Analóg videó kimeneten jeleníti meg a szükséges adatokat és a spektrumot.

2.1. Gamma spektrum

A mért sugárzás energiáját X , sűrűségét/gyakoriságát pedig Y tengelyen ábrázoljuk. Az így kapott spektrum csúcsai mutatják meg a mért térre jellemző gamma sugarakat (1. ábra). Minden radioaktív izotóp egy sajátos spektrummal rendelkezik.

Gyakori példa a Cs137 csúcsa 662keV energiánál:



1. ábra: Gamma spektrum

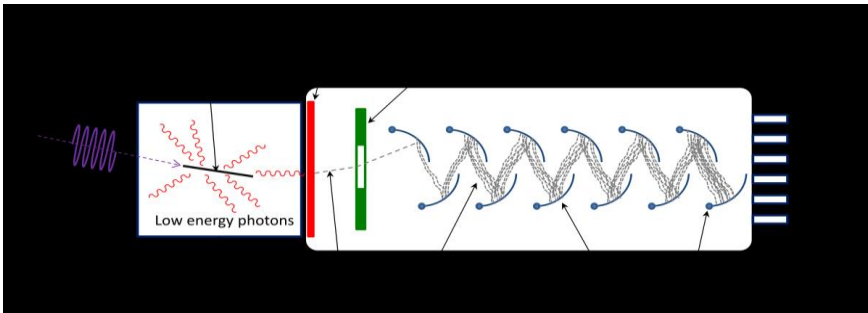
2.2. NaI(Tl) Szcintillátor



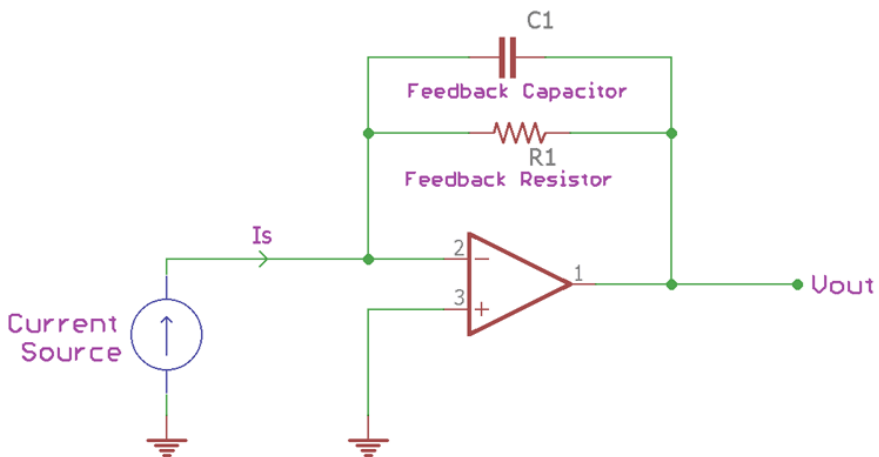
2. ábra: Szcintillátor kristály

A kép egy extrém esetet mutat (2. ábra), a fényhatás általában nem látható szabad szemmel. Az erősítésre használt cső (PMT) a fotoelektromos hatáson alapul, így a katódról leütött elektronok lavina formájában haladnak az anód felé (3. ábra). Ez a kimeneti jel.

A további feldolgozásra alkalmas áramkör (transzimpedancia erősítő) (4. ábra)



3. ábra: Detektor



4. ábra: Transzimpedancia erősítő

$$\text{Erősítés: } V_{out} = -(I_s \times R1) \quad (1)$$

2.3. Mikrovezérlő

Miután a PMT jelei tovább lettek erősítve (1. képlet), a kimeneti feszültséget egy Atmega 2560 mikrovezérlő digitalizálja és tárolja. Az egyszerűbb megjelenésű értékekért (pl.: mért sugarak sűrűsége, üzemmód) egy kisebb kijelző felel. A spektrum és a komplexebb kezelői felület pedig analóg NTSC/PAL videójelként továbbítja (labori körülményekhez alkalmas) megjelenítésre az adatokat¹.

Erre a kijelző típusra ezen okokból esett választás:

¹Ingyenes könyvtár: <https://code.google.com/archive/p/arduino-tvout/downloads>

- Egyetlen vezeték szálon történik a kommunikáció, így egyszerű a bekötése.
- Ingyenes könyvtár van hozzárendelve.
- 128×96 felbontást biztosít.
- Külső átalakító modullal bármilyen kijelzőn elfut a rendszer.
- Esztétikailag retro/sajátos hatást ad.

2.4. Program

A fentebb említett felhasználói felület alkalmassá teszi az eszköz személyre szabhatóságát (pl.: hangbeállítások, figyelmeztetések, jegyzetek). Minden hosszútávon érvényes adatot (például a spektrum is) egy külső EEPROM fog tárolni, csatlakozva a mikrovezérlőhöz, így ezek az adatok a kikapcsolást követően is megmaradnak. A programozás Arduino-n keresztül történik mivel egyszerű és széles eszköztárt kínál mikrovezérlők és könyvtárak szempontjából is.

Főbb funkciók:

- save data (eeprom mentés)
- hang beállítások (hangszín, némítás, alarm/figyelmeztető)
- flag system (a spektrumhoz és pontjaihoz flagek/zászlók tűzhetők, egyedi jegyzetekkel)
- üzemmódok (gamma intenzitás, gamma spektrum)
- téma (változatos ikonok, esztétikus megjelenítés)

3. Elért eredmények:

Eddigi eredménynek számít az alap áramkör megvalósítása, mérési eredmények fogadása és feldolgozása, saját programozású menü demo és EEPROM kezelő a detektor összeszerelése és erősítése.

Telemetry Systems

Kandó Racing Team

Kovács Tamás, Váradi Marcell, Kiss Csaba

Felkészítő tanár: Ladányi Sándor

Kecskeméti SZC Kandó Kálmán Technikum, 6000 Kecskemét, Bethlen körút 63.

1. Bevezetés

Az iskolánkban informatika és elektronika szakon tanulunk. Mégis több hardware fejlesztése inspirált bennünket, hogy létrehozzuk a Telemetry Systems-et. Ötletünk az adatok gyűjtésére és vizualizálására fókuszál, egy webes felületen keresztül, ami számos hasznos alkalmazási lehetőséget kínál. Például raktároknál és üvegházaknál a páratartalom és hőmérséklet kijelzésére, uszodákban a medencék hőmérsékletének mérésére, gyárakban az elkészített termékek számának ellenőrzésére vagy akár egy parkolóban a szabad férőhelyek kiírására. Célunk, hogy mindezt jobban és elérhetőbb áron tudjuk megvalósítani, mint más hasonló, sok esetben régi és elavult, rendszer a piacon.

2. Probléma megoldásának menete

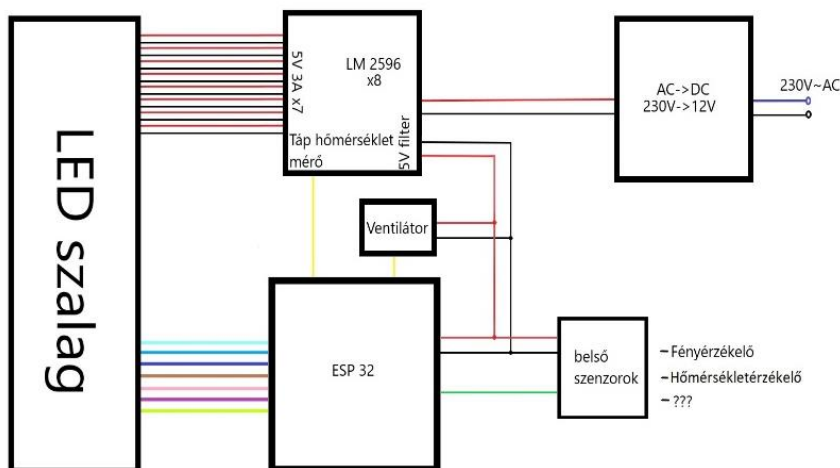
Eredeti ötletünk az iskolánk parkolójának modernizálása volt, csak később, a hardware megépítése után jöttünk rá, hogy ebből a projektből többet is ki lehetne hozni. A távoli elérés egy html oldalon keresztül lehetővé teszi az automatizálás és távfelügyelet funkcióinak bővítését.

2.1. Elektronika



1. ábra: LED kijelző

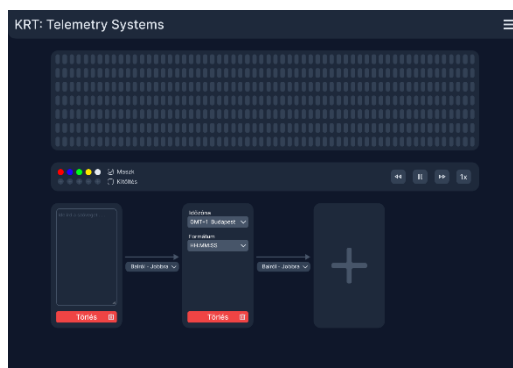
A kijelző tervezését és megépítését Csabi (elektro.) intézte, ő tervezte meg a dobozát és a működtető áramkörét. Az áramkör tervezéséhez Sprint Layout tervező programot, a 3D modellekhez Fusion 360 CAD programot használt. A kijelző vezérlése egy NodeMCU ESP32S WiFi-s mikrovezérlő, ez irányítja a 7-szer 1 méteres WS2816B led szalagot és kommunikál az online felülettel. Az 1. ábrán látható a kijelző első tesztelése, a "Petőfi 200" szöveggel. Iskolánk márciusban emlékezik meg nagy költőnkéről és itt minden szakmánk készül valamivel, mi ezt a szöveget íratjuk ki az aulában a 420 pixeles legújabb fejlesztésünkkel. A piros-fehér-zöld virtuális zászlónk csak feldobja az animációs szöveget.



2. ábra: Felépítési terv

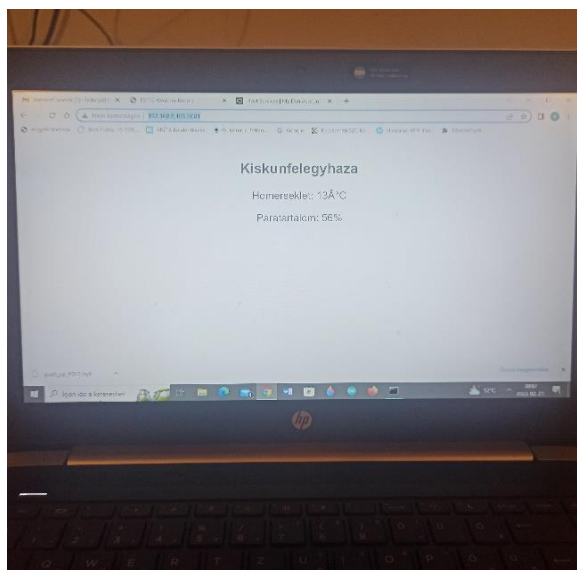
2.2. Szoftver

Az ESP programozásához VSCode-ot használtuk a Platform IO bővítménnyel. A backend fejlesztésért nagyrészt Tamás volt a felelős. Ez C++ programozási nyelvben került megvalósításra és a csillagpontos hálózat elvein alapszik. Marci a webes felülettel és a UX-el foglalkozott, ehhez a Svelte JavaScript keretrendszert használta. A nyílt forráskódú szoftver elérhető mindenki számára és részletes dokumentációnak köszönhetően nem kell szakember a beállításához.



3. ábra: Webes felület a kijelző irányítására

A külső mérési adatokat gyűjtő szerverek telepítésével hőmérséklet és páratartalom értékeket kezdtünk el gyűjteni. Az alábbi képeken a telefonon és számítógépen megjelenített mérési adatok láthatóak.

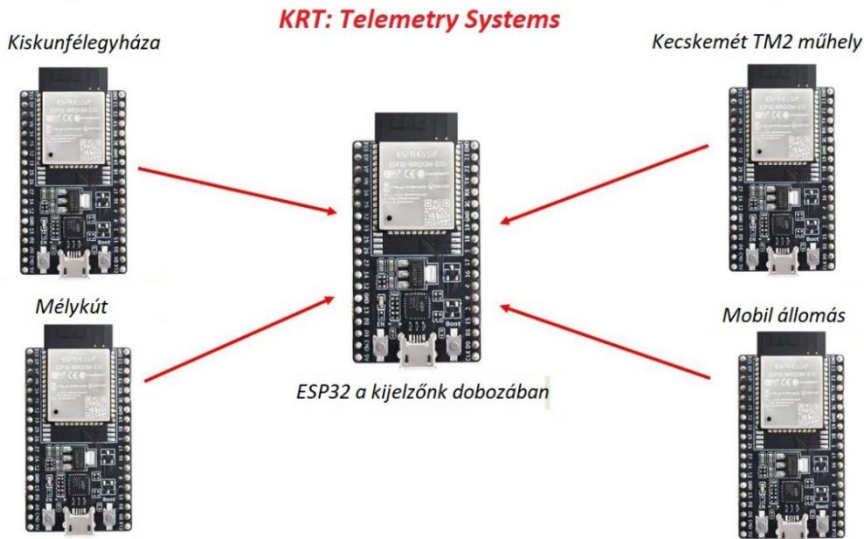


4. ábra: Mérési adatok

3. Elért eredmények

Eredeti terveinkhez képest ötletünk nagyon kinőtte magát. Ami csak egy iskolai okos parkoló projektként indult, már egy teljes körű, sok felhasználásra alkalmas, telemetria rendszer. 5db ESP32-es vezérlőt használtunk egy hálózat építéséhez. Az első adatgyűjtő állomásunkat Kiskunfelegyházán állítottuk fel, a másodikat Kecskeméten a sulink innovatív

tanműhelyében indítottuk el. Még egyet Mélykútra juttatunk el, és egy mobil mérőállomás is készült a termékeink bemutatásához. Az 5. vezérlés a kijelzőnk dobozában kapott helyet. Már nem csak szövegeket tudunk kiíratni, hanem több szervertől mérési adatokat is gyűjtünk és a webes felületen is megjelenítjük.



5. ábra: A telemetria rendszer

Nyílt forráskódunknak köszönhetően bárki elkészítheti a saját verzióját és nagyon költséghatékonyan használhatja a rendszerünket adatok gyűjtéséhez és azok megjelenítéséhez, bármilyen formában. Köszönjük, hogy elolvasta! Tisztelettel a KRT: Telemetry Systems csapata!

Optimális billentyűkiosztás tervezése a magyar nyelvhez

Fizikus módszer

Kovács János

Felkészítő tanár: Sikó Dezső

Kecskeméti Bolyai János Gimnázium, 6000 Kecskemét, Irinyi utca 49.

1. Bevezetés

Egy problémára adott első megoldás mindig hatalmas előnyre tesz szert a későbbiekkel szemben, pedig az utána levők sokszor nagyságrendekkel jobbak. A billentyűzetkiosztásokkal sincs ez másképp. A QWERTY-t mindenki ismeri, de kevesen vannak azzal tisztában, hogy mennyire nem effektív. Számos alternatív kiosztás terjedt már el angol nyelvterületen, a legismertebbek közülük a DVORAK és a COLEMAK. Egyéb nyelvekre sokkal kevesebb alternatív kiosztást terveznek, melyek gyakran csak a DVORAK kiegészítései ékezetes betűkkel, ami nem veszi figyelembe egy adott nyelv sajátosságait.

2. Probléma megoldásának menete

Projektemben számos szempont figyelembevétel egy új kiosztást dolgoztam ki a magyar nyelvre.

Először kutatást végeztem, hogy milyen szempontok szerint szokás billentyűzetet tervezni, illetve saját megfigyeléseket is végeztem.

Főbb szempontok:

- Effort Layout és betűgyakoriság $\Leftrightarrow$ A billentyűzet egyes billentyűit nem ugyanolyan nehéz leütni, a könnyebben elérhetőkre igyekeztem helyezni a gyakoribb betűket. Az 1. ábrán a Colemak projekt effort layoutja látható, a 2.-on pedig a saját preferenciám alapján módosított verzió.
- Munka optimális megosztása a kezek között: tapasztalatok alapján az az optimális, ha a domináns kéz végzi a leütések 55%-át, ez 90%-ban a jobb kéz.
- Alternáló vagy roll leütések maximalizálása, illetve optimális arány keresése
- A hagyományos billentyűzeten a gépelés nagy része az alsó és a felső soron történik, így a gépírónak folyamatosan változtatnia kell ujjai helyzetén, ami a gépelést lassabbá és fárasztóbbá teszi. Ezért igyekeztem a gyakoribb betűket az alapsorra helyezni.
- Arra törekedtem, hogy az egyes ujjak igénybevétele arányos legyen azok erősségével
- Azonos kézzel végzett kényelmetlen leütések minimalizálása

- o Same Finger Bigramok 2% alatt tartása: Same finger bigramnak nevezik az azonos ujjal végzett különböző billentyűk egymás utáni leütését. Ezek lassítják a gépelést, ezért igyekszünk ritkává tenni őket.
- o Azonos kézzel, de különböző ujjakkal alsó és felső soron végzett egymás utáni kényelmetlen leütések („scissors”) minimalizálása.
- o Azonos kézzel, különböző ujjakkal végzett távoli billentyűk egymás utáni leütésének minimalizálása.

A magyar nyelv számos tulajdonságában tér el az angoltól. Más a betűk gyakorisága, a gyakori betűkombinációk is. Nyelvünk kerüli a mássalhangzótorlódást, csupán az „sz” és „cs” párosok gyakoriak, illetve a magánhangzók is ritkán szerepelnek egymás mellett, szemben az angol „ou”, „ea” kombinációkkal. A magyar nyelv jóval több féle magánhangzót használ, viszont a „w”, „x” és „q” betűket alig. Szükségem volt a magyar nyelvben a betűk gyakoriságának ismeretére, erre készítettem egy saját szövegelemző programot. A kapott eredményeket összevettem egy nagyobb adatgyűjtéssel is, látható, hogy az eredmények nagyjából megegyeznek, ha nem is pontról pontra, de hasonló gyakoriságú betűcsoportok ugyanabban a sorrendben szerepelnek.

Szépirodalmi szövegek (saját elemzés):

E A T L N S O K R I Z M G É Á Y D V H B J Ö F U P Ó C Ő Ü Í Ú Ű

5000 cikk alapján Magyarország leglátogatottabb gazdasági hírportáljáról (interneten talált elemzés):

E A T L S N K I Z R O Á É G M B D Y V P H J Ö U F Ó Ő C Í Ü Ú Ű

4 millió magyar mondatból álló magyar korpusz (saját elemzés):

E A T L S N K I R Z O Á É G M Y B D V H J P U Ö Ó F C Ő Í Ü Ú Ű

A magyar bigramok gyakoriságáról nem találtam részletes forrást, úgyhogy elkészítettem saját elemzésem. Az eredmény a 3. ábrán látható.

Amint az várható volt, kiderült, hogy a magyar nyelvben a mássalhangzó-mássalhangzó bigramok aránya jelentősen kisebb, mint az angolban, melynek adatai a 4. ábrán láthatóak.

Az egy kézre eső bigramokat érdemes olyan billentyűpárookra helyezni, melyeket könnyű gyorsan, egymás után leütni, és nem ugyanarra az ujjra esnek.

Saját billentyűzetkiosztásom az 5. ábrán látható. Első ránézésre feltűnik, hogy a magánhangzók a bal, a mássalhangzók a jobb oldalra vannak szervezve. Így az alternáló leütések aránya eléri a 83%-ot. A Same Finger Bigramok aránya csupán 1,59%. A jobb kézzel végzett leütések aránya az optimális 55%. Gépelés során a mutató és középső ujjak dolgoznak legtöbbit. A leütések több mint fele az alapsoron történik, a felső és alsó sort jóval kevésbé használja a kiosztás, a számsort pedig szinte egyáltalán nem, a w, x és q betűk alig fordulnak elő magyar nyelvű szövegben.

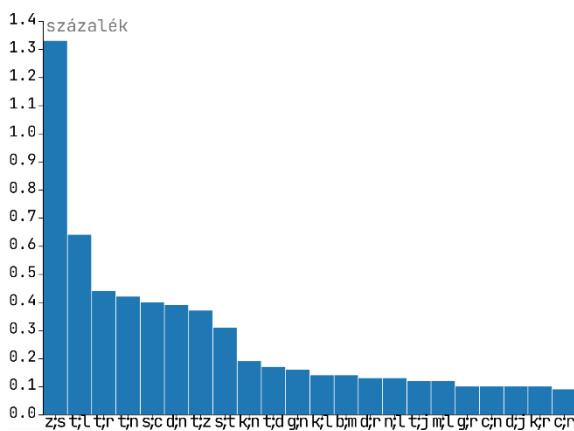
2.1. Ábrák



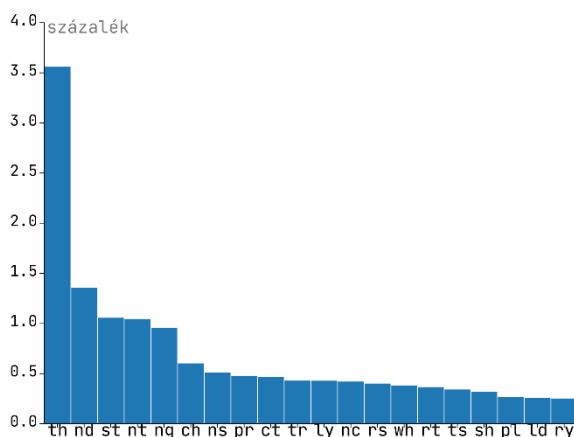
1. ábra: Effort layout a Colemak projektből

Tab	8	5	3	4	7,5	9	4	3	5	8	10		Enter
Cap	2	2	1	1	7	7	1	1	2	2	11		
Shft	8	6	4	3	5	12	5	3	4	6	8,5		Shft
Control		Alt									Alt		Control

2. ábra: Saját preferenciám alapján módosított effort layout



3. ábra: mássalhangzó bigramok a magyar nyelvben (párosok a betűket mindkét sorrendben tartalmazzák, azaz a „z;s” jelenti a zs és sz betűt is)



4. ábra: mássalhangzó bigramok az angol nyelvben (párosok csak a felírás sorrendjében, azaz a „th” oszlop nem tartalmazza a „ht”-t

0	'	"	+	!	%	/	=	(	)	w	x	q	Backspace
Tab	ó	u	á	y	í	f	z	k	d	h	ú	ű	Enter
Cap	i	o	e	a	ő	b	l	t	s	n	p	-	
Shif	ü	.	,	é	ö	c	v	r	m	g	j		Shif
Control	Alt	[F12]										Alt	Control

5. ábra: saját billentyűzetkiosztásom

3. Elért eredmények

Ezt a billentyűzetkiosztást használom mindenre a QWERTY helyett, amikor tanórákon jegyzeteket készítek, programozok, ennek a leírásnak a gépelésére is. Eddig elért maximális sebességem 133 szó/perc

Rendszeresen kísérletezek, hogyan lehetne a legtöbbet kihozni egy billentyűzetből magyar szövegbevitel terén. Nem feltétlenül kell minden ékezetes magánhangzónak saját billentyű.

Terveim közt szerepel egy új billentyűzet tervezése, melyben nagyobb hangsúlyt fektetnék a roll leütésekre, egyéb szempontokra, illetve alkalmaznék valamilyen új koncepciót az ékezetes betűk bevitelére.

Egy optimális billentyűzet szabványosítása számos előnnyel járna. A gépírók képzési ideje csökkenne, nagyobb maximális sebességet érnének el, az irodai munkát végzők kezeinek megterheléséből eredő megbetegedések is csökkenhetnének.

A projekt korábban bemutatásra került a Kutató Diákok Mozgalma Műszaki- és reáltudományi Tematikus Konferenciáján. (Mivel ez egy olyan verseny, ahol felvételi pontokat is lehet nyerni, egy műszaki projektet feltétlenül benevez rá egy középiskolás.)

Hordozható szigetüzemű napelemes MPPT rendszer kifejlesztése

SolarWise

Varga Bence, Jordán Nikolasz

Felkészítő tanár: Kovács Livia

*Közgazdasági Politechnikum Alternatív Gimnázium,
1096 Budapest, Vendel utca 3.*

1. Bevezetés

Szeretnénk létrehozni egy kompakt, hordozható, szigetüzemű napelemes rendszert, mely egyúttal a lehető legkisebb környezeti lábnyommal rendelkezik. A rendszer lényege, hogy egy különálló, mobilis, okos akkumulátor pakkot lehessen szabadon használni, valamint csatlakoztatni a hordozható napelem panelekhez. Az eszköz fel lesz szerelve többféle kimenettel; telefon-, laptop- vagy bármilyen USB-s eszköz töltését támogatni fogja, mindemellett egy 230V-os AC invertert is szeretnénk elhelyezni rajta. Általános hétköznapi felhasználásra készül, kempingezéshez, hétvégi házakba, garázsba.

2. Probléma megoldásának menete

2.1. Céljaink

Célunk, hogy megtervezzünk és saját magunk legyártsunk egy működő prototípust. Mindemellett szeretnénk bemutatni egy olcsó, praktikus, mindenki által használható, környezetbarát szigetüzemű napelemes rendszert.

2.2. Felépítés

Egy házban helyezük el az akkumulátort, a vezérlőegységet és a kimeneteket. Ez a kompakt rendszer egy 3D nyomtatott, könnyű, ám mégis ellenálló és környezetbarát műanyag (PLA) házban kap helyet. A házon elhelyezett ergonomikus fogantyúk könnyebbé teszik a szállítást.

A napelem paneleket különálló, multifunkcionális rögzítőpontokkal látjuk el, így minden felületre rögzíthetők lesznek. Ennek, és a beépített vízmértékeknek köszönhetően a napelemek dőlésszögét mindig ideálusra állíthatjuk be.

2.3. Napelemek

Az akkumulátor pakk töltését két, egyenként 60 W-os, 20 V-os napelem fogja biztosítani. Ezek együtt 120 W-os töltési teljesítményt tesznek lehetővé, mellyel a rendszer körülbelül 5 óra alatt teljesen feltölthető lesz napsütéses időben. A maradék rendelkezésre álló napenergia is felhasználható lesz kimeneteken keresztül.

2.4. Akkumulátor pakk

Egy legalább 500 Wh kapacitású Li-ion akkumulátor pakkot szeretnénk építeni. Az ehhez szükséges 18650-es típusú akkumulátor cellákat rossz laptop- és szerszámgép akkumulátorokból bontjuk ki. Ezzel rengeteg kidobásra ítélt akkumulátort mentünk meg és hasznosítjuk őket újra. Tesztelés után a legjobb cellákat kiválogatva egy biztonságos és strapabíró akkumulátor pakkot építünk belőlük. Az egységeket ponthegeztéssel rögzítjük egymáshoz, és egy 35 A terhelhetőségű BMS-sel védjük meg a pakkot az esetleges elektromos hibáktól.

2.5. Vezérlés és kijelzés

A rendszer szívéét két darab 16 MHz-es Arduino Nano mikrokontroller fogja alkotni, melyek egymással kommunikálni és kollaborálni fognak. Az egyik mikrokontroller a feszültség-, áramerősség- és hőmérséklet méréséért lesz felelős, illetve a saját fejlesztésű MPPT (Maximum Power Point Tracking) rendszert fogja üzemeltetni. Ennek lényege, hogy a napelemből a lehető legnagyobb teljesítményt nyerve ki az áramkör, a körülményektől függetlenül. Emellett a másik Arduino egy 1.8 colos színes LCD kijelzőt és a kimeneteket fogja vezérelni. A kijelző adatokat, információkat és üzeneteket jelenít meg, így intuitívabbá téve a rendszert. A kimeneteket külön-külön MOSFET tranzisztorok fogják kapcsolni, ezeket gombok segítségével lehet vezérelni. A teljesítmény tranzisztoros vezérlés nagy előnye, hogy igen jó hatásfokú, még nagy teljesítmény esetén is.

2.6. Felhasználás

A kompakt házba, az akkumulátor mellé szeretnénk beépíteni minél több csatlakozót, modult, mellyel a multifunkcionalitást növeljük. Megtalálható lesz az eszközön 2 db 5V 2A-es USB-A csatlakozó, 2 db 60 W-os, USB-C laptop töltésére is alkalmas gyorstöltő csatlakozó, valamint egy vezeték nélküli töltő is helyet kap. Mindemellett beépítünk egy 12 V 10A terhelhetőségű szivargyújtó aljzatot, és egy legalább 500 W-os 230 V AC invertert, mellyel a használható eszközök tárháza a végtelenre bővül. A hálózati csatlakozónak köszönhetően bármilyen speciális eszköz használható és tölthető, amelyre egy hétévi háznál vagy kempingezésnél szükség lehet. Emellett a biztonságra is kiemelt figyelmet fordítunk, ahol csak lehet biztosítékokat, védődiódákat és egy főkapcsolót is beépítünk. Az Arduino programozásánál is nagy hangsúlyt fektetünk a beépített komponensek védelmére. Ha a mikrokontrollerek bármi rendellenességet észlelnek a mért adatokban, azonnal leoldanak, ezzel megvédve az eszközt.

3. Elért eredmények

Büszkék vagyunk, hogy az Országos Tudományos és Innovációs Olimpián (OTIO 2023) az összesen 157 beküldött pályamunkából bizalmat szavaztak nekünk, és továbbjutottunk a második fordulóba. A Közgazdasági

Politechnikum Tudományos és Innovációs Tehetségkutató Pályázatán szintén bekerültünk a második fordulóba. Az akkumulátorpakk ponthelesztéséhez egy saját, kompakt ponthelesztőt is készítettünk. A napelemes rendszer kifejlesztésében is számos téren léptünk előre az utóbbi két hónap során. Saját, színes LCD kijelzős UI-rendszert (1. ábra, balra) programoztunk és építettünk meg, melyet gombok segítségével lehet vezérelni. A kijelzőt vezérlő Arduino az információ- és adatkijelzésen kívül a kimenetek kapcsolásáért is felelős lesz. Az ehhez szükséges saját tervezésű MOSFET teljesítmény-tranzisztoros kimenetvezérlést is (1. ábra, balra), elkészítettük és teszteltük. Jelenleg a két napelem panel karakterisztikáját és működését vizsgáljuk (2. ábra), így hatékonyabb MPPT rendszert (1. ábra, jobbra) fejleszthetünk ki.



1. ábra: LCD kijelző és kimenetek (balra), MPPT töltésvezérlő (jobbra) vizsgálata oszcilloszkóppal



2. ábra: A két, új napelem panel első tesztelése mesterséges fény alatt

4. Társadalmi haszna

Ez a projekt nem csupán a technikai megvalósításról szól, hanem a környezetvédelemről és a szigetüzemű rendszerek népszerűsítéséről is. Célunk, hogy mindenkihez eljusson a megtérülő, olcsó és zöld elektromos áram. Mindemellett sok helyen nem megoldható az elektromos hálózat kiépítése, melyre kiváló megoldást kínál az általunk fejlesztett kompakt eszköz.

5. Környezetvédelem

Kívülről a legszembetűnőbb, hogy minden műanyag alkatrész biológiailag lebomló PLA műanyagból, 3D nyomtatott technológiával készül. Befelé haladva azonban még több környezetvédelmi és költségcsökkentő megoldással találkozunk. Az akkumulátor pakk 100%-ban bontott, bevizsgált, jóminőségű 18650-es Li-ion cellákból épül fel. Az elektronikai alkatrészek egy része szintén bontott, használt, de jó minőségű. Az elektronikai rendszert a lehető legjobb hatásfokúra tervezzük, így minimalizálva az energiaveszteségeket. Ráadásul az eszköz könnyen szerelhető, alkatrészei cserélhetők, így garantálva a hosszú élettartamot. A napelem panelek is kiváló minőségű, hosszú élettartamú darabok, így sok évig gond nélkül használhatóak majd. A környezetbarát legyártás mellett maga a napenergia használata járul hozzá ahhoz, hogy minimalizáljuk környezeti lábnyomunkat.

6. Jövőbeli terveink

A hordozható szigetüzemű napelemes rendszerünkről szeretnénk készíteni oktató jellegű videókat, melyekkel bemutatjuk az eszközt, ezzel is terjesztve a környezettudatosságot, fenntarthatóságot.

Tisztább Iskolát!

Robbineers

Bulyák Balázs, Kertész Levente, Soós Tamás

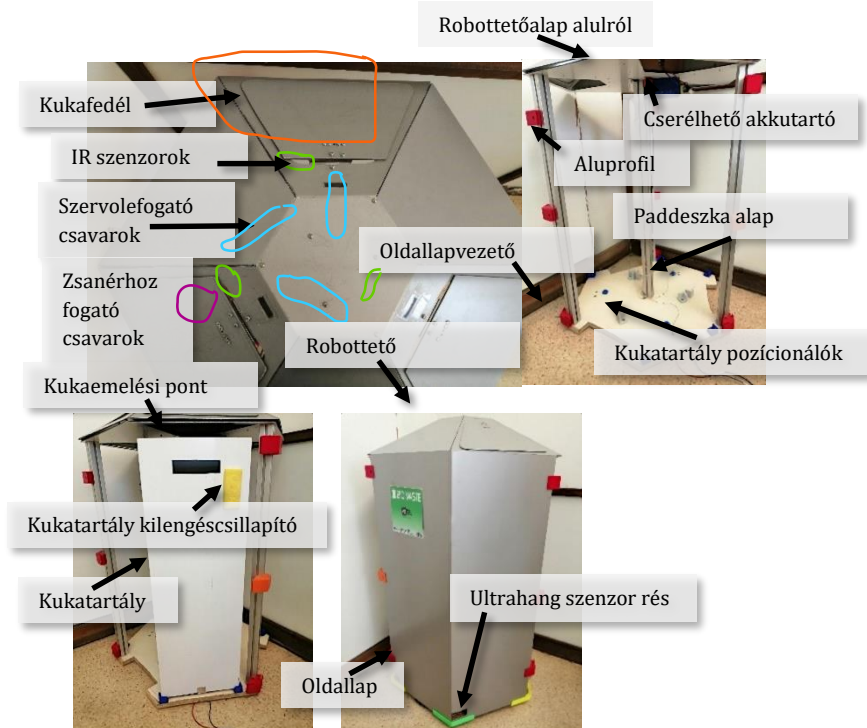
Felkészítő tanár: Nové Lajos, Kóra Gábor, Nagymáthé Tamás

BGSZC Mechatronikai Technikum, 1118 Budapest, Rétköz utca 39.

1. Bevezetés

Manapság komoly problémát jelent a személtelés. A diákság körében ez sajnos egy elterjedt szokás, amely sok problémát okoz. Ezért a mi iskolánk is célul tűzte ki, hogy megpróbálja csökkenteni a személtelés mértékét egy jobban elérhető kukával.

Az ötletünk az volt, hogy építünk egy önjáró szelektív hulladékgyűjtő robotot, amely járja az iskola folyosóit ezzel emlékeztetve a diákokat arra, hogy a megfelelő helyre dobják ki a szeméjtüket. Most már megtehetik ezt akár



1. ábra: A kukarobot szerkezete a valóságban

szelektíven is - a folyosón - ennek a robotnak a segítségével. A megvalósítás ránk, diákokra várt, amelyet bár tanári, szülői segítséggel, de főként mi magunk igyekeztünk megoldani.

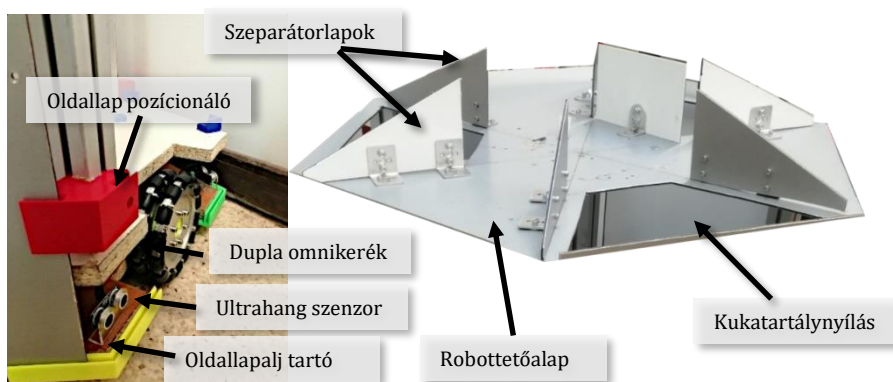
2. Probléma megoldásának menete

A feladat kivitelezését három fő részre bontva mutatjuk be:

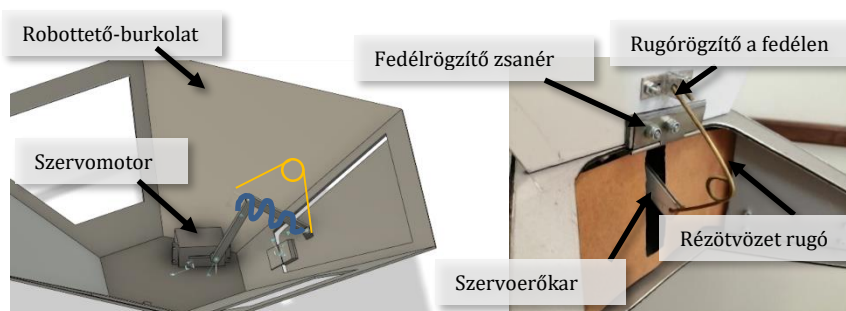
2.1. Robot szerkezeti felépítése, mechanizmusai

A robot szerkezetét valós alapon fotókkal, illetve egy modellen keresztül prezentáljuk az 1., 2. és a 3. ábrákon.

Rugó szerepe: Kukafedélfelfeszítés/lecsapásnál a mozgatószerkezet nem törik el, senkinek sem tudja odacsukni a kezét.



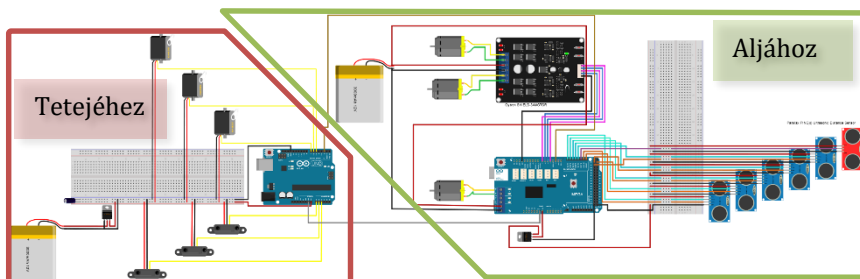
2. ábra: A kukarobot szerkezete a valóságban



3. ábra: Kukafedélnyitás mechanizmusa

2.2. Robotban használt elektronika

A robot elektronikáját a 4. ábrán látható módon valósítottuk meg. A robotban használt alkatrészek darabszámát, megnevezésüket az 1. táblázatban ismertetjük.



4. ábra: A robot kapcsolási rajza

Elektronikai alkatrészek	
Mennyiség (db)	Megnevezés
1	Elektrolit kondenzátor (10 μ F)
1	Cytron SHIELD-3AMOTOR motorvezérlő
5	HC-SR04 ultrahangos távolságérzékelő szenzor
2	5V DC feszültszabályozó
3	SHARP GP2Y0A21YK IR közelítésérzékelő szenzor
3	FITEC FS6530M szervomotor
3	MODELCRAFT 7127C DC motor
1	Parallax PING))) ultrahangos távolságérzékelő szenzor
1	Arduino Mega 2560 (Rev3)
1	Arduino Uno (Rev3)
1	Arduino Motor Shield (Rev3)
2	NiMH 3000mAh akkumulátor 12V

1. táblázat: Kukarobot elektronikai alkatrészei

2.3. Robothoz tartozó program

Robot működési elve, programok:

A robotot jelenleg két Arduino program együttese irányítja. Ahogy azt a 3. ábráról, illetve az 1. táblázatból is ki lehet találni, az egyik program az Arduino Unora, a másik az Arduino Megára van feltöltve. Az Unora feltöltött program irányítja a kukafedelelek nyitását, csukását. A Mega-

program vezérli a robot mozgását. Továbbá az Uno jelzi a Mega felé minden kukafedélnyitást annak érdekében, hogy megállítsa a robotot miközben a szemetet dobják ki.

Programkód-részlet:

Kukafedélnyitás/csukás:

```
if (pos1 == 40 and dis_cm1 > 10) { //Ha a szervo
    40 fokban van és az IR-en mért táv nagyobb, mint 10 cm
    while (pos1 > 0) { //Addig ismétli, amíg a          szervo 0
        fokot el nem éri
        digitalWrite(megaPin, HIGH); //Mega STOP!
        pos1-=1; //Elvesz egy fokot a pos1 változó értékéből
        myservol.write(pos1); //Ráírja a pos1 változó      értékét a
        szervora
        delay(50); //Vár 50ms-et
    }
} else {
    digitalWrite(megaPin, LOW); //Mega mehet
    pos1 = pos1; //Tartja a szervo pozícióját
    myservol.write(pos1); //Ráírja a szervora a      pozíciót
}

if (dis_cm1 <= 10) { //Ha az IR táv kisebb, mint 10 cm
    if (pos1 == 0) { //Ha a szervo 0 fokban áll
        while (pos1 < 40) { //Addig csinálja, amíg  kisebb a
            pozíciója, mint 40 fok
            digitalWrite(megaPin, HIGH); //Mega STOP!
            pos1+=2; //Növeli a ciklusváltozó értékét          2-vel
            myservol.write(pos1); //Ráírja a szervora
            delay(50); //Ismét vár 50ms-et
        }
        delay(2000); //Fenntartja a fedelet 2mp-ig
    } else {
        digitalWrite(megaPin, LOW); //Mega mehet
        pos1 = pos1; //Tartja a pozíciót
        myservol.write(pos1); //Ráírja a szervora
    }
}
```

3. Elért eredmények

Sikerült egy tökéletesen működő béta verziót összehoznunk. A robotunk már tud:

- önállóan mozogni az ultrahangszenzorok segítségével
- korrektül nyitni a kukafedeleket
- még meg is áll a szemétkidobás idejére

Tervben továbbá:

- veszélyes helyek kiküszöbölése IR-es vonalérzékelővel
- gyorsulásmérős ütközésselhárítás (fennmaradó holtterek miatt)
- Raspberryvel meghajtott érintőképernyő

TGAM projekt

TGAM

Vadász Balázs

Felkészítő tanár: Dr. Farkas József, Várady Zoltán

BMSZC Neumann János Informatikai Technikum, 1144 Budapest, Kerepesi út 124.

1. Bevezetés

Az alapötlet abból a feltételezésből indul ki, hogy az emberi agy frekvencia-tartományokba rendezve értelmezi a szenzorjai felől beérkező adatokat, és azokat bioelektromágneses hullámként dolgozza fel. A szakirodalomban sokféleképpen írják le a feldolgozó mechanizmust. A mi elméletünk ezek alapján a következő:

Tudjuk, hogy a beérkező jelek előfeldolgozása a ingereket felfogó szenzoroknál történik, melyek a mechanikai, kémiai és hullámtermészetű ingereket bio-elektromágneses rezgésekké alakítják. Ezek a rezgések hullám formájában áradnak szét a különböző agyi képletek felé. A folyamat a fejbőr meghatározott pontjain elhelyezett érzékelőkkel felfogható és elvezethető egy hozzá illesztett feldolgozóhoz. Elképzelésünk szerint erre kiválóan alkalmas a TGAM chip, amely a NeuroSky elsődleges agyhullám-érzékelő ASIC modulja. A chip egy ThinkGear nevű protokollt használ a kommunikációra. Ezt a protokollt a Neurosky cég fejlesztette ki és az API-ját ThinkGear Serial Stream SDK csomagjában elérhetővé tette. Maximális fogyasztása 15 mA. Eredetileg tömegpiaci alkalmazásokhoz tervezték, elsősorban játékos alkalmazások támogatására. A modul feldolgozza a kimenetek EEG frekvencia spektrumát.

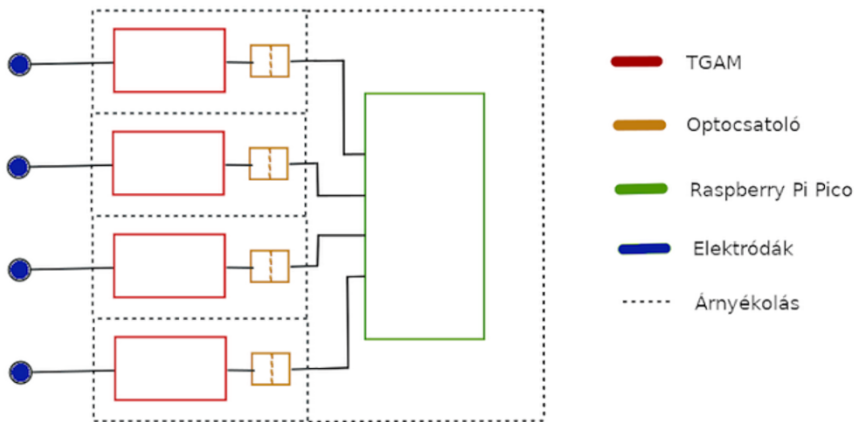
Az elektroenkefalográfia (EEG) egy pszichofiziológiai mérőeljárás, melynek segítségével az agyban zajló bio-elektromágneses hullámok keletkezését, illetve az agy működésének élettani hátterét vizsgálhatjuk, szűkebb értelemben pedig egy elektrofiziológiai mérőeszköz, mely az idegsejtek elektromos aktivitásának regisztrálására szolgál. Az EEG-vel elvezethető jel az elektroenkefalogram, amely egy komplex, több komponensű periodikus görbeként írható le. Az EEG technológia hat kimutatható hullámtípust jelenít meg, amplitúdójuk és frekvenciájuk különbözősége alapján (alfa, Mu, béta, delta, gamma, théta). Kutatások kimutatták, hogy egy-egy információ frekvencia-specifikus jellel digitálisan is leírható.

A TGAM a nyers EEG-jelből a figyelem, a meditáció és a pislogás által keltett agyhullámokra specializálódott, de mert az érzékelt hullámok teljes spektruma belefér a chip érzékenységi tartományába, alkalmasnak véljük az agy ébrenléti állapotában keletkezett hullámainak érzékelésére, mérésére.

Egyszerű száraz elektródákkal ez a modul kiválóan használható felhasználási területtől függetlenül mérési célokra, mert hordozható és könnyen üzembe helyezhető.

2. Probléma megoldásának menete

Az általunk kifejlesztett mérőeszköz négy TGAM chip rendszerbe illesztésével, valamint 4 optocsatoló bekötésével egy Raspberry Pi (RPi) Pico típusú célszámítógéphez juttatja el az adatokat (1. ábra). A chippek mindegyike egy speciális ötvözetből készült száraz elektródát használ, melyeket a koponyára kell rögzíteni. Ezen kívül két fülre csíptethető bemenetre van szükség, egy végtelen potenciálú testre (árnyékolás) és egy referencia-vezetékre (a fülön és a koponya fejbőrén elhelyezett érzékelők közötti feszültségkülönbség kimutatására).



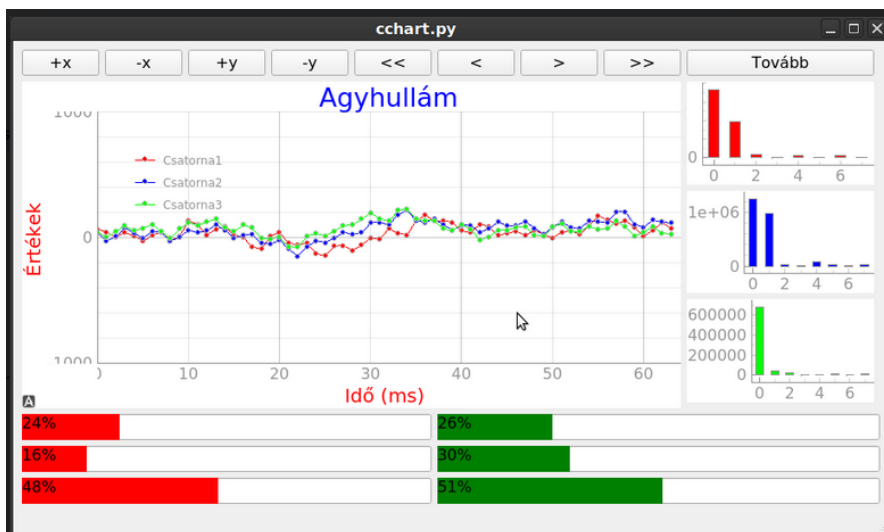
1. ábra: 4 csatornás agyhullám-olvasó sematikus ábrája

A chip lelke egy ASIC (alkalmazásspecifikus integrált áramkör), amely a másodpercenkénti adatokat nyolc frekvenciasávra bontja, illetve kiszámolja a figyelmi és relaxációs értékeket százalékban kifejezve. A nyers adatokat, a számolt értékeket és egyéb információkat, mint például a jel/zaj viszony szintje, egy aszinkron soros vonalon keresztül továbbítja. A tesztelés folyamán azt tapasztaltuk, hogy az eszköz nagyon érzékeny a külső elektromágneses zajra, ezért egy teljesen árnyékolt formát terveztünk, illetve a soros kimeneteit chipenként egy optocsatolóval választottuk le a feldolgozó egységtől. Az RPi bemenetére sorosan érkező jeleket az optocsatolók analóg formában szűrik és erősítik. A hordozhatóság fenntartása miatt a négy chip jelét egy MCU egységgel dolgozzuk fel. A rugalmas táplálhatóság, illetve alacsony fogyasztás és a Wifi támogatás miatt a Raspberry Pi Pico-ra esett a választásunk. A Pico hardveres támogatással, akár nyolc soros Tx vonalat és két full duplex soros vonalat tud működtetni. Az eszköz tápellátására egy 3,7 V-os 900 mA-es

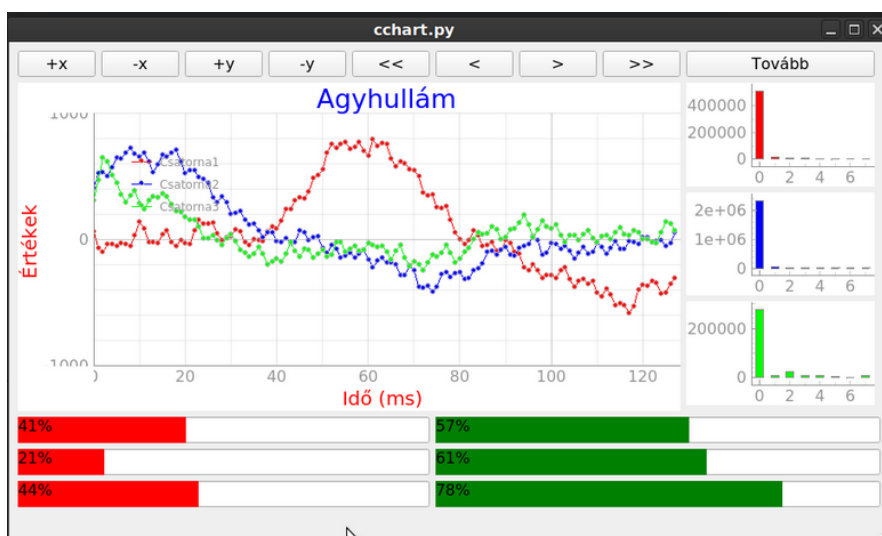
akkumulátort használunk, amellyel akár több, mint nyolc órás folyamatos üzemelés is megvalósítható. Az RPi digitalizálja az adatokat 512 Hz-es mintavételezéssel. A Pico-ban futó program egy saját fejlesztésű keretrendszert használ a négy csatorna szétválasztására és időbélyegekkel való ellátására. Ez a program szerver üzemmódban fut és a Wifi eszköz titkosítását használva látja el a PC-én futó kliens programot az adatokkal. A PC-én futó klienst Python nyelven írtuk meg, feladata a ThinkGear protokoll megvalósítása és az adatok megfelelő megjelenítése és feldolgozása.

3. Elért eredmények

A TGAM chipekből érkező négy hullám egyetlen monitorképpel megjeleníthető. A tervezett eszköz prototípusa egy 3 csatornás rendszer volt, amelynek monitorképei az alábbiakban láthatók (2. és 3. ábra):



2. ábra: A 3-csatornás TGAM monitorképe működő agyról (relaxációs agyhullámok)



3. ábra: A 3-csatornás TGAM monitorképe működő agyról (megzavart aktivitású agyhullámok)

Ezután a kapott adatok kiértékelhetők és analizálhatók. Az eredményeket több területen is hasznosítani lehet. Például különböző neurológiai betegségek (Alzheimer, demencia, Parkinson stb.) korai stádiumban kimutathatók. Továbbá egyes afáziás kórképek diagnosztizálásában támogatást nyújthat (jellemző minták keresésével – mesterséges intelligenciával).

A továbbfejlesztési lehetőségek hosszú távon az említett betegségek gyógyítását is elősegíthetik. Háromszögeléses technikával meg lehetne határozni például az agyban keletkezett sérülések, léziók pontos helyét.

Testkulcspontok becslése FMCW radarképek segítségével

vajdad4m

Vajda Ádám

Felkészítő tanár: Dr. Vajda Róbert

ELTE Radnóti Miklós Gimnázium, 1146 Budapest, Cházár András u. 10.

1. Bevezetés

1.1. Célkitűzés

Kutatási célom egy olyan mesterséges intelligencia modell megtervezése és betanítása volt, amely képes valós időben, vizuális okklúzióban felismerni egy személy testének kulcspontjait radarképek alapján.

1.2. Téma aktualitása

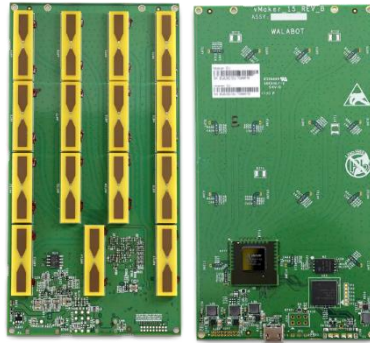
A mesterséges intelligencia taktikai és aszimmetrikus előnyt biztosíthat a közelharcban. Az emberi test kulcspontjainak (a továbbiakban: póz) becslése a számítógépes látás egyik sokat kutatott területe. Többek között a megfigyelésben, a gesztusfelismerésben és a virtuális valósággal való interakciókban használható. Hasznos információkat szolgáltatathat többek között a természeti katasztrófák utáni mentési műveletek során is.

Számos olyan helyzet van, amikor a hagyományos, kamera alapú pózfelismerés nem alkalmazható a látási viszonyok, vagy esetleg a kilátást akadályozó akadály miatt. A radaralapú pózfelismerés viszont megoldja ezt a problémát. Ugyanakkor szükségünk van egy olyan módszerre, amellyel a póz adatokat kinyerhetjük a radarképekből.

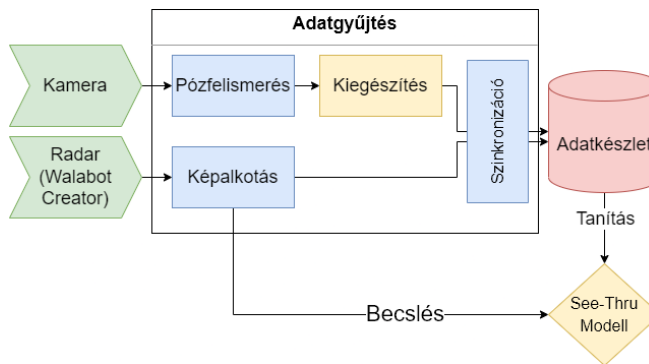
1.3. Megvalósítási terv

Az FMCW radarok hatékonynak bizonyulnak a kis hatótávolságú képalkotás megoldásában. Ebből a célból a Walabot „Creator” radarját választottam.

A kutatás megvalósításához saját radar- és póz adatokat kellett gyűjtenem, amihez egy radar-kamera szinkronizációs rendszert fejlesztettem. Kerestem egy ígéretes módszert a póz kulcspontjainak a kameraképekből való kinyerésére. Ki kellett dolgoznom egy modellt, hogy kiegészítsem a hiányzó kulcspontokat. Mindezek segítségével pedig egy olyan modellt terveztem és tanítottam fel, amely a radarképek alapján kiszámítja a kulcspontokat.



1. ábra: Walabot „Creator” Radar



2. ábra: Megvalósítási terv

2. Probléma megoldásának menete

2.1. Radar

A radar paramétereit a kamera paramétereinek (90°-os látószög, 640×480-as felbontás) figyelembevételével optimalizáltam. Látóterének 60°-ot választottam, hogy megakadályozzam a határon túli méréseket. Ismert, hogy a pózfelismerő módszer elveszti pontosságát, ha a célpont a képkocka szélén van, ezért a pózfelismerést, így a radart a képkocka belső részére kell fókuszálni. Ez a 15°-os különbség mindkét irányban, úgy tűnik, hogy megoldotta ezt a problémát.

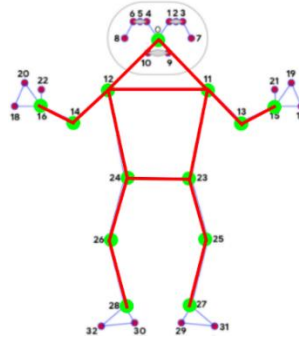
2.1. Kamera és pózfelismerés

A testkulcspontok felismeréséhez a Google BlazePose (Bazarevsky, et al., 2020) módszerét használtam, amely 640×480-as méretű képekből 32 egyedi kulcspontot képes felismerni és követni. Ebből a 32 kulcspontból 13

	R	θ	ϕ
Res / Felbontás	5 cm	1°	1°
Min	100 cm	-30°	-30°
Max	250 cm	30°	30°

2. táblázat: A radar mérési paramétereit

főkulcsponthot választottam ki követésre és mentésre, a többi 19-et elvetettem. Ez azért szükséges, mert a radar felbontása nagyobb, mint a törölt pontok közötti távolság (például ujjak közötti távolság), ezért teljesen felesleges számítási kapacitást igényelnének.



4. ábra: Kiválasztott kulcsponthok

2.2. Pózkiegészítés

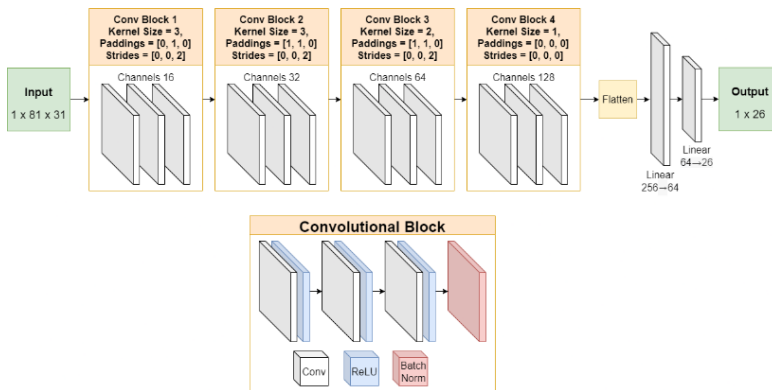
El kell ismerni, a fent említett módszerrel történő rögzítéskor nem minden kulcsponth megállapítható. Ennek a problémának a megoldására egy olyan modellt használtam, amit teljesen rögzített kulcsponthokon képeztem ki. A modell veszteségfüggvénye az alábbi:

$$L = \sum_{i=1}^n \sqrt{[640(R_{x_i} - P_{x_i})]^2 + [480(R_{y_i} - P_{y_i})]^2}$$

A veszteségfüggvény az euklideszi távolságok pixeltéri összege a megjósolt és a címkézett kulcsponthoktól mérve. L a kapott veszteség értéke, R_x , illetve R_y a valós póz x és y koordinátáinak vektora, P_x és P_y , pedig a becsült póz koordinátáinak vektora. A pózkiegészítő modell architektúrája egy 0,2-es meredekségű Leaky ReLU aktivációs funkció MLP modell.

2.3. Pózkinyerés radarképekből

Az adatgyűjtést elvégezve ~1700 póz-radarkép pár állt rendelkezésemre, ezek különböző helyzeteket, illetve különböző cselekvéseket (sétálás, ülés, állás) rögzítő mérések. A radarképekből a test kulcspontjait kinyerő modell egy konvolúciós neurális hálózat, amelynek két lineáris réteg a feje. A CNN-ek használatának számos előnye van. Először is, jól működnek a képekkel kapcsolatos feladatokban, másodsor pedig, kevés paramétert igényelnek. A modell összesen 1,7 millió paramétert tartalmaz, ami gyors következtetési időt eredményez. Az architektúrája egy VGG-16 (Simonyan & Zisserman, 2014)-hoz hasonló modell, módosított kimeneti fejjel.



5. ábra: A modell architektúrája

3. Elért eredmények

Az eredmények azt mutatják, hogy kulcspontonként átlagosan **7,5 cm**-en belüli pontosságot érhetünk el. Megfigyelhető, hogy ezt a feladatot egy embernek szinte lehetetlen lenne elvégezni, de egy mesterséges intelligencia modell képes nagy pontossággal végrehajtani. A radarképek világosan mutatják a koncentrációt, ahol az ember áll. Ez alapján a személy radartól való távolsága is mérhető.

4. Összefoglalás

Munkám során bemutattam egy olyan rendszert, amely képes az emberi test tájékozódási pontjainak felismerésére radarképekből. Ennek megvalósításához két mély tanulási modellt hoztam létre. Az egyik képes nagy pontossággal megjósolni a hiányzó emberi kulcspontokat egy adott pózban. A másik modell 13 különböző kulcspontot tud reprezentálni a radarképekről.

A bemutatott rendszer alkalmazható terrorelhárításnál, természeti katasztrófák utáni mentési műveleteknél, repülőgépek biztonsági szintjének növelésére, gesztusvezérlésben és egyéb biztonsági rendszerek fejlesztésében.

Arduino-vezérelt automatikus öntözőrendszer

Péter Attila

Péter Attila

Felkészítő tanár: Luca Krisztián

Zalaegerszegi SZC Széchenyi István Technikum, 8900 Zalaegerszeg, Déryné utca 1.

1. Bevezetés

A növények nagyon kevés gondozást igényelnek, és napokig is felügyelet nélkül hagyhatók, viszont amikor egy-két hétnél hosszabb utazásaink alatt nem tudjuk gondozni a növényeket a talaj nedvességhiánya miatt károsodhatnak. Ilyen helyzetekben a növény elszorvhat vagy elpusztulhat a megfelelő öntözés hiánya miatt. Erre a problémára találtam megoldást amikor kitaláltam az Arduino Uno-val ellátott és vezérelt automata öntözőrendszert, amely automatikusan öntözi a növényeket, és akkor is egészségesen tartja őket, ha hetekig vagy hónapokig nem tudunk velük foglalkozni. Ebben a projektben egy nedvességérzékelőt használok a növények optimális nedvességszintjének fenntartásához.

A rendszer egy Arduino mikrokontrollert használ az öntözési folyamat automatizálására, ami hatékonyabbá és pontosabbá teszi azt. Programozható úgy, hogy bizonyos időpontokban vagy a talaj nedvességszintje alapján kapcsoljon be és ki. Ez nem csak precíz vízgazdálkodást tesz lehetővé, amely időt és erőforrásokat takarít meg, de segíthet megelőzni a túl- vagy alul öntözést is, ami egészségesebb növényeket eredményez. Ezzel a rendszerrel kézi beavatkozás nélkül biztosíthatjuk, hogy növényeink a megfelelő időben megfelelő mennyiségű vizet kapjanak.

A következőkben a rendszer alapváltozatát készítem el, ami tökéletesen ellátja az öntözési feladatokat egy vagy több növény esetében. A rendszer többféleképpen továbbfejleszthető, így akár üvegházakban, kertekben, farmokon is használható lehet. A rendszer teljesen automatikussá tehető úgy, hogy amikor a vízszint-érzékelő a vízszint alacsony állásáról figyelmeztet, akkor egy, a vízhálózathoz csatlakoztatott szivattyút egy relé vagy áramlásvezérlő segítségével működésbe hozunk, ami addig tölti a tartályt, míg a víz szintje nem érte el a kívánt értéket. Mindennek tökéletes alapja ez a vizsgamunka.

1.1. A rendszer működésének rövid leírása

Az Arduino szolgál a rendszer agyaként. Fogadja az érzékelőktől érkező jeleket, feldolgozza a beérkező adatokat és azoktól függően vezérli a rendszert. Az érzékelők, például a talajnedvesség-érzékelők, valamint a vízszint-érzékelő információkat gyűjtenek a környezet aktuális viszonyairól. Ezeket az információkat használjuk fel annak meghatározására, hogy mikor és mennyit kell öntözni a növényeket.

A szivattyúkat az Arduino vezérli. A rendszert úgy programoztam, hogy az meghatározott időközönként vagy bizonyos feltételek – jelen esetben a talaj nedvességtartalma alapján – kapcsolja be a szivattyúkat. A rendszer ellenőrzi a talaj nedvességét, így az, ha alacsony bekapcsolja a szivattyút, ami megöntözi a növényt, majd 3 másodperc múlva kikapcsol. Ha a talaj még mindig nem elég nedves, újra indul a folyamat mindaddig, amíg a növény megfelelő mennyiségű vizet kapott.

A rendszer áramforrása kettő akkumulátor. Az egyik áramforrás biztosítja az Arduino kártya és az érzékelők működéséhez szükséges energiát, a másik pedig a szivattyúkat működteti.

A rendszer az Arduino programozási nyelv és az Arduino IDE (Integrated Development Environment) segítségével programozható. A kód megírható úgy, hogy olyan konkrét paramétereket tartalmazzon, mint például a kiengedendő víz mennyisége, az öntözés közötti időintervallumok és a szivattyúk aktiválásának feltételei. Az öntözést az akkumulátoron található kapcsolóval ki- és be lehet kapcsolni. A rendszer állapotát egy kijelzőn olvashatjuk, ami jelzi a talaj jelenlegi nedvességét, valamint azt, hogy a víztározó tartályban van-e elegendő víz az öntözéshez.

2. Probléma megoldásának menete

Az összeszerelést a sorkapocs és az Arduino rögzítésével kezdtem (1. ábra). A sorkapocsra ezután rákötöttem az Arduino 5V-os, illetve GND lábát. A pozitív a kék, a negatív pedig a fehér színű lett.

Ezek után a szivattyúk működtetésére használt reléket kötöttem be. A relék VCC lábait rákötöttem az sorkapocs pozitív felére, a GND lábait a földre, a jel lábait pedig az Arduino 7-es, illetve 6-os digitális lábaira.

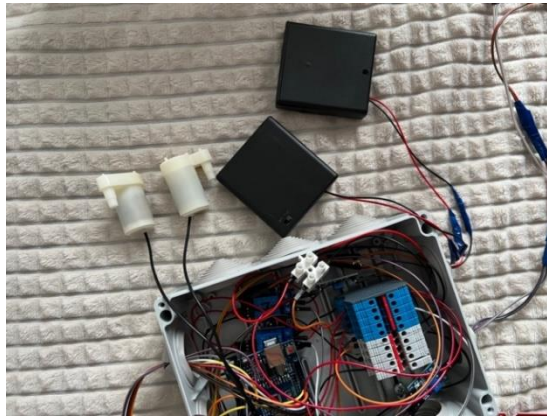


1. ábra: A rögzített Arduino

Következő lépésként a két talajnedvesség-érzékelő bekötésével folytattam. Csatlakoztattam az érzékelők VCC lábait az 5V-ra, a GND lábait a

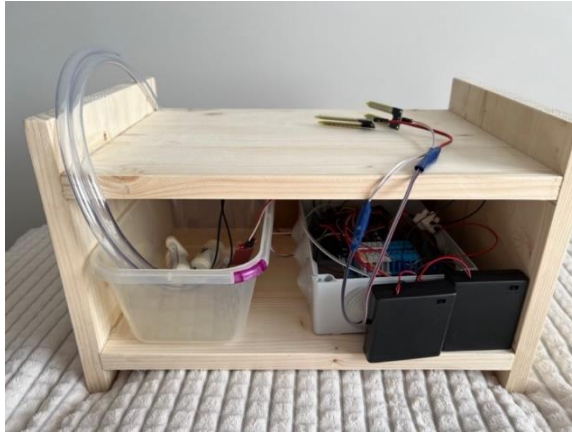
földre. Mivel én az analóg értéket használtam fel a programban, így az érzékelők analóg kimeneti lábait kötöttem az Arduino A0-ás és A1-es lábaira.

Az összeszerelést a szivattyúk bekötésével folytattam. A szivattyúk negatívját összeforrasztottam és csatlakoztattam a földre. A szivattyúkat, a reléket és a talajnedvesség-érzékelőket is 1-es, illetve 2-es számmal jelöltem meg, hogy tudjam melyik eszköz, melyikhez tartozik. Az 1-es szivattyú pozitívját az 1-es relé, a 2-es szivattyú pozitívját a 2-es relé NO (alaphelyzetben nyitott) lábára kötöttem. Csatlakoztattam a rendszer működéséhez szükséges áramforrást is. A rendszer tesztelése során egy olyan problémára derült fény, hogy az Arduino, a szivattyúk bekapcsolását követően lefagyott, nem működött tovább megfelelően. A szivattyúk nem kapcsoltak ki, és a program további részei sem futottak le. Mivel az egész rendszer egy, 4 db, 2,5V-os elemekből álló akkumulátorról működött, arra gondoltam, hogy szivattyúk működésbe lépését követő, hirtelen feszültségesés miatt, az Arduino kártyának nem maradt elég áram a megfelelő működéshez. Ennek megoldására a reléket és a szivattyúkat egy másik akkumulátorról kezdtem működtetni, ami az elvárásaimnak megfelelően megoldotta a problémát. (2. ábra)



2. ábra: Akkumulátorok

Ezután csatlakoztattam a vízszint-érzékelőt. pozitív lábát az 5V-ra, negatív lábát a földre, az analóg lábát pedig az A2-es analóg bemenetre. Csatlakoztattam még egy mini hangszórót is, (buzzer) ami a víztározó tartály kiürülésekor szaggatott hangjelzéssel figyelmezteti a felhasználót, hogy a rendszer nem működik. Ezután csatlakoztattam a kijelzőt is. Az összeszerelés során figyelembe vettem a cseréptartó kialakítását, ennek megfelelően vezettem ki az alkatrészeket a dobozból, hogy azokat megfelelően tudjam elhelyezni (3. ábra).



3. ábra: Cseréptartó

3. Elért eredmények

Az elkészült rendszer tökéletesen ellátja kettő növény automatikus öntözését, ezzel egészségesen tartva őket. Apró módosításokkal akár több növényt is öntözhetünk vele, csak egy új szivattyú, relé és a program bővítése kell hozzá. Az Arduino vezérelt öntözőrendszer számos előnnyel rendelkezik a hagyományos öntözőrendszerekkel szemben többek között:

1. Automatizálás: Egy Arduino által vezérelt öntözőrendszer programozható úgy, hogy meghatározott időpontokban és ideig automatikusan öntözze a növényeket, biztosítva, hogy a növények megfelelő mennyiségű vizet kapjanak kézi beavatkozás nélkül.
2. Testre szabás: A rendszer testre szabható, hogy megfeleljen a különböző növények sajátos öntözési igényeinek. Például egyes növények gyakoribb öntözést igényelnek, mint mások, és az Arduino programozható az öntözés gyakoriságának és időtartamának megfelelően.
3. Víztakarékoság: Az Arduino által vezérelt öntözőrendszer a talaj nedvességszintjének érzékelésére szolgáló szenzorok segítségével biztosítja, hogy a növények csak szükség esetén kapjanak vizet, megakadályozva a vízpazarlást.
4. Költséghatékony: Az Arduino kártyák és érzékelők viszonylag olcsók, így az Arduino által vezérelt öntözőrendszer költséghatékony megoldás kis- és nagyszabású rendszerekhez egyaránt.

Tehát, az Arduino által vezérelt öntözőrendszer hasznos, mivel testre szabható, automatizált és költséghatékony megoldást kínál a növények öntözésére, miközben vizet és energiát takarít meg.

Informatika szekció

Családfa-számítás

SWP

Török Péter, Kocsis Nándor, Domokos Ádám

Felkészítő tanár: Bíró Zsolt

SZTE Gyakorló Gimnázium és Általános Iskola

6722 Szeged, Szentháromság utca 2.

1. Bevezetés

Csapatunk célja egy olyan rendszer létrehozása volt, amely képes genetikai számítások elvégzésére, valamint a számítás eredményének átlátható megjelenítésére. Ezt egy olyan felülettel értük el, ahol a felhasználó egy családfát hoz létre, majd egyetlen gomb megnyomásával tudja elvégezni a számításokat.

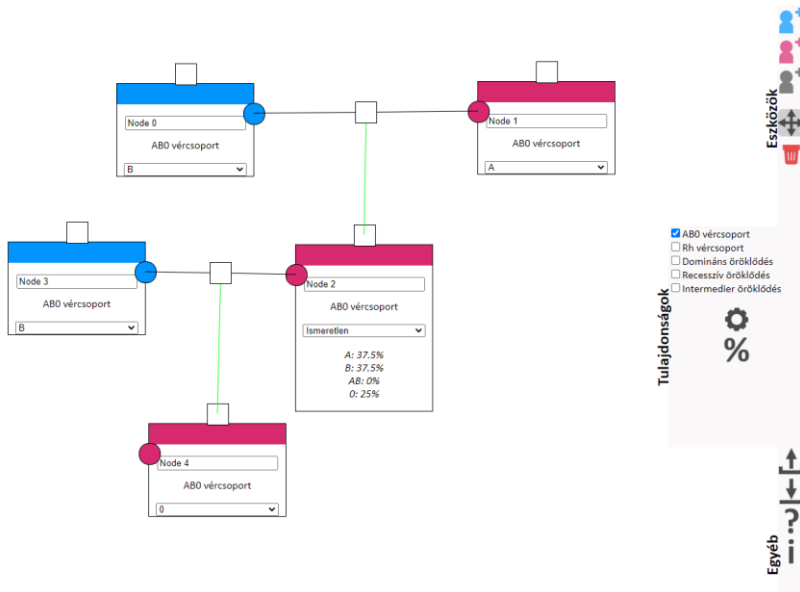
2. Probléma megoldásának menete

A széleskörű elérhetőség érdekében egy weboldal készítése tűnt a legjobb megoldásnak. Ehhez a HTML, a CSS valamint a JavaScript nyelveket használtuk. A projekt két fő részre osztható; a kezelőfelületre - amelynek részét képezi a családfaszerkesztő és a tulajdonságszerkesztő -, valamint a genetikai számításokat végző programra.

2.1. A kezelőfelület

A kezelőfelület leglényegesebb elemei a személyeket jelző Node-ok, ugyanis ezek összekapcsolásával lehet elkészíteni a családfát. Ezek olyan div tag-ek, amelyeket a felhasználó szabadon mozgathat. A háttérben egy canvas segítségével rajzolódnak ki a személyeket összekötő vonalak. Ezt a canvas-t a Node-ok pozíciójától függően át kell méretezni, hogy bárhol látszódjanak a vonalak. A kezelőfelület az 1. ábrán látható.

A kezelőfelület létrehozása során a tulajdonságszerkesztő elkészítése vett igénybe hosszabb időt. A felhasználó itt tudja megadni az egyes genotípusokhoz tartozó fenotípusokat. Ezt a felületet program a megnyitásakor, a meglévő adatok alapján generálja le, majd bezáráskor a felhasználó által beállított adatokat tölti vissza a rendszerbe. Ennek eredményeként mentéskor és betöltéskor nem kell külön kezelni a tulajdonságok adatait.



1. ábra: Kezelőfelület

2.2. A genetikai számításokat végző program

A projekt genetikai számításokat végző része képes nagy körben becslések elvégzésére. Képes sok családtagból álló családfák feldolgozására: nem csak az utódok de a szülők tulajdonságainak megjóslására is, lyukas családfák kitöltésére, kettőnél több allélt tartalmazó gének feldolgozására, nemhez kötött öröklődésre stb. Így sok genetikai tulajdonság és betegség öröklődését nyomon lehet vele követni vagy visszakeresni. A futási idő kényelmes marad nagyjából hat tagig, de ez fejlesztve lesz a közeljövőben. A program támogatja az időutazást is.

3. Elért eredmények

A genetikai számítások helyesen működnek, valamint felhasználóbarát felületet sikerült biztosítani, amellyel könnyen végezhetőek el a számítások. A program felhasználható öröklött tulajdonságok (például vércsoport) vagy betegségek előrejelzésére, orvosi célból genetikai betegségek nyomon követésére, vagy oktatási célokra.

A jövőben figyelmet szeretnénk fordítani súgó létrehozására, amely megkönnyíti a használatot a genetikában nem jártasok számára is, mert a tulajdonságszerkesztő ilyen szempontból még kihívást jelenthet egyeseknek. Az elkészült projekt [github](https://swpgithubofficial.github.io/Csaladfa-szamitas/)-on² megtekinthető.

² <https://swpgithubofficial.github.io/Csaladfa-szamitas/>

A világ legelső Brainfuck operációs rendszere (FirstOS)

Asszem... bili

Bábel Edvárd István

Felkészítő tanár: Harmath Zsolt

Móra Ferenc Gimnázium, 6100 Kiskunfélegyháza, Kossuth Lajos u. 9.

1. Bevezetés

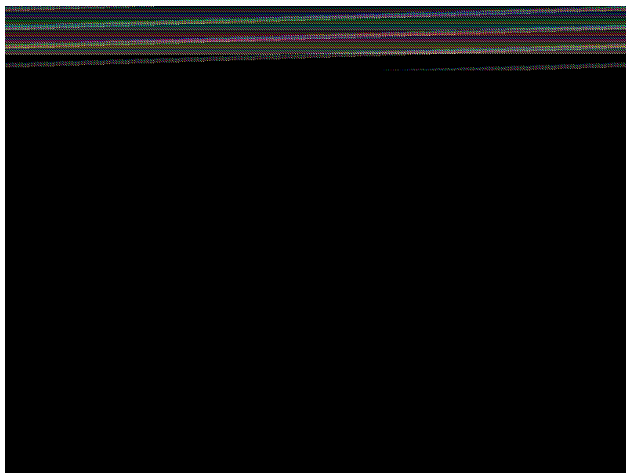
Ezt még akkor találtam ki, amikor rátaláltam a Brainfuck nevű nyelvre. Első látásra megszerettem. A neve, használati módja pont hozzám passzol, nekem is ilyen kaotikus a gondolkodásom ezért relatív gyorsan tudok benne programozni. Értelme van, hisz brainfuckot lehetetlen feltörni, hihetetlen gyorsan fut, kevés adatot foglal. Csak hát kell valami segédprogram, hogy lehessen valamit a képernyővel is csinálni. Bár az LFB-t (Linear Frame Buffer-t) és a fájlrendszert még be kell fejezni, szerencsére, nem egy hatalmas munka. Internet-kompatibilisre tervezem, és hardver függetlenre. Most igaz, x64-bites (NASM) Assembly az alapja, tehát ezen kívül másik distro-n nem fut, de ha a Windows és a Linux meg tudta csinálni, nekem is sikerülni fog. De ezek nem lesznek sajnos könnyűek.

2. Problémák, hogyan működik

Elkezdtem rákérteni, hogy hogyan lenne lehetséges ezt megcsinálni, És hát mondjuk úgy, hihetetlen messzire jutottam. Minden, ami működött, saját magam írtam. Az Interpreter-t, és még egy saját programozási nyelvet is csináltam közben. Liner-nek neveztem el mivel a karakterek, amiket használni kell ebből állnak: l, I, 1, !, és persze a 0, hogy az alappontba visszaugorhassak. Tervezek még hozzá rakni parancsokat, ami pl. A képernyő közepére viszi a mutatót stb. Funkcionalitásért. Kissé nehéz elmagyarázni mivel annak ellenére, hogy milyen kicsi, hihetetlen komplex. A nagyon alap programozás részével nem volt gondom, valamiért nekem mindig a nehezebb részei könnyebbek. Viszont a grafikára gondolva fel áll a szőr a hátamon. Bizonyos videó kártyák máshogy működnek, ki kell számítani, hogy mi a címe a bájtoknak, amit át akarunk írni, hogy lássunk valamit. Aztán még a Linear Frame Buffernek is kész kell lennie, hogy az egész képernyőt is lehessen használni. És még rá tesz egy lapáttal, hogy ezt át kell alakítani Binárisba, majd virtuális gépen kell futtatni. És átfutottunk az 512 bájtton, ami azt jelenti, több szektort be kell olvasni a merevlemezről. Őszintén ez az életem legnehezebb projektje, és már hónapok óta csinálom. Ez a hihetetlen sok sornyi kód az eddigi életem legnagyobb műve. Ráadás képp, hogy egyedül csináltam.

2.1. Ábra

Itt lehet látni csodásan, ahogy egy mozdulattal elrontottam a képernyőmet, (1. ábra) és csak ronda szörnyűségeket írt ki. Végül szerencsére rájöttem, hogy miért volt, rossz volt a számolás...



1. ábra: A videó-mód nincs optimalizálva.

2.2. Az egyenlet

Ez az az egyenlet, ami megváltoztatta a program menetét. (1. egyenlet) Nem kellett végre szenvednem, beírogatni egyesével a címet, hanem csak automatikusan rájött a számítógép magától, hogy mit is kéne csinálnia.

$$\text{cím} = (X + Y * \text{Felbontás}) * 3 \quad (1)$$

2.3. Táblázat

Itt vizuálisan lehet látni egy táblázaton hogyan, működik a Liner. Akármilyen nyelvhez hozzá lehet rakni, csak kapcsos zárójelek közé kell írni a programot.

Lehet látni, hogy a parancsok hova viszik a „mutatót” ahol beszínezi a képernyőt. (Relatív parancsok a 0-pozíciótól. Brainfuck alapon működik.)

Képernyő	
0	1
l	1l

1. táblázat: Liner működése

3. Elért eredmények

- Nagyon kevés tárhelyet foglal a számítógépen.
- Hihetetlenül gyorsan fut.
- Hibátlanul működik az operációs rendszer.
- Nem tud kékhalált kapni, maximum lefagy, vagy lefagy.
- Nem lehet meghackelni, mivel visszafelé lehetetlen feltörni (Assembly → Brainfuck) nem lehet bejutni a brainfuck Interpreterbe, mert assemblyben lesz megprogramozva az internet rá, minden fontosabb futtatóparancsokkal együtt. Legrosszabb esetben is a rendszer maximum lefagy.
- Olyan alapszinten működik, hogy a processzor hatalmas hatékonysággal fut. És ráadásképpen csak akkor ha pl: megnyitunk egy fájlt. Semmi féle program nem lassítja a háttérben. Teljes hatalom a számítógép felett.
- Biztonságosan lehet tárolni adatokat.
- Multi-funkciós, csak rá kell tölteni egy Brainfuck → más nyelv Interpretert (pl. java, hogy minecrafterhozassunk rajta).

A fájlrendszere teljesen különleges. Elmondani nagyon szívesen elmondanám, de ezt egy zárt OP-rendszernek tervezem, hogy feltörhetetlen maradjon. És, legalább majd nem tudják lecrackelni. A rendszert őszintén mindenki fogja tudni használni, bár azok, akik értenek a programozáshoz; vagy arra használják, azoknak sokkal jobb ötlet használni. És cégeknek is, hisz az adatokat, amíg nem töltenek le semmi vírusos dolgot és van proxyjuk (Proxy FirstOS) technikailag lehetetlen, hogy feltörjék őket.

Arcfelismerő alkalmazás fejlesztése jelenlét ellenőrző rendszerhez

TeachHelper

Csizmadia Ádám, Szalai Áron

Felkészítő tanár: Szabó Beatrix, Balogh Attila

*Szegedi SZC Tóth János Mórakalmi Szakképző Iskola és Szilágyi Mihály Kollégium,
6782 Mórakalom, Dosztig köz 3.*

1. Bevezetés

A tanulmányaink során, valamint az általunk részt vett órákon azt tapasztaljuk, hogy egyre több időt vesz el egy-egy órából az, hogy a tanár ellenőrizze az órán jelen levő diákok névsorát, mivel a tanulói jelenlét ellenőrzésére a mai napig a papír alapú adminisztráció a legelterjedtebb, ugyanis a tanár nem csak a Kréta rendszerben, hanem a saját jegyzeteiben is vezeti, hogy ki van jelen a tanórákon. Kutatási témánkkal erre a problémára szeretnénk megoldást nyújtani, hiszen egy olyan szoftvert szeretnénk fejleszteni, mely rövid időn belül fel tudja ismerni a teremben jelen levő személyeket és ezzel megkönnyíti a tanulók órán való nyilvántartását.

2. Probléma megoldásának menete

Kutatásunk nagyobb részét a szoftverek fejlesztése illetve a megfelelő algoritmus implementálása teszi ki. Az arc felismerésre rendelkezésre állnak kész, működőképes algoritmusok. Viszont ezeknek az algoritmusoknak az alkalmazása is komoly kihívásokkal néz szembe, mert számolni kell a különböző környezeti hatásokkal, amelyek kihatnak a számolás pontosságára, mint például: terem nagysága, a megvilágítás mértéke, a fejét a padon pihentető hallgató, a padon hagyott hátizsák, a hallgatók mozgása, a kamera felbontása. Több kutatás is rámutatott arra, hogy az órán való részvétel és a diák tanulási teljesítménye között szoros az összefüggés. A projekt célja pontosabb értékeket szolgáltatni a tanulók létszámáról, és csökkenteni az esélyt arra, hogy a rendszerben olyan diákok is szerepeljenek, akik nincsennek jelen az órán.

A projekt működéséhez szükséges hardver elemek: számítógép, kamera.

Felhasznált programozási nyelvek a fejlesztéshez:

- Asztali alkalmazás: C#, MariaDB az adatok tárolásához, json az adatcseréhez, mesterséges intelligencia (OpenCV) az arcfelismeréshez.

3. Elért eredmények

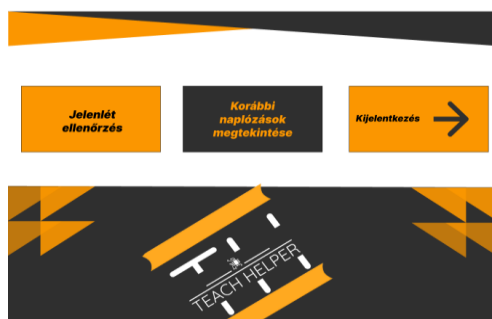
Mivel a rendszerünk még nem rendelkezik admin résszel, így a felhasználók (tanárok) első lépésként regisztrálnak, majd pedig bejelentkeznek a felületre.

A regisztrációs ablakon (1. ábra) a tanár meg kell, hogy adja a teljes nevét, felhasználónevét, email címét, és jelszavát. Regisztráció során egy algoritmus összehasonlítja a táblában lévő adatokat és a beírt adatokat ha, létezik már a felhasználó, akkor nem engedélyezett a regisztráció megvalósítása. Amennyiben a felhasználó már szerepel a regisztrációt tartalmazó táblában, akkor a regisztrált adatok megadásával lehetőség van a bejelentkezésre (2. ábra). Ha a felhasználónév és a jelszó nem egyezik meg a regisztráció során megadottal, akkor a beírt adatok törlődnek és a felhasználó újra próbálkozhat a bejelentkezéssel. Ha az adatok egyeznek, akkor tovább lép a felhasználó az feladatválasztó felületre.

1. ábra: Regisztrációs ablak

2. ábra: Bejelentkezés ablak

A bejelentkezés után a következő oldal, amely megjelenik a tanár számára az egy feladatválasztós felület, ahol a tanár eldöntheti, hogy tovább megy a jelenlét ellenőrzésre vagy megtekinti a már naplózott órákat. Valamint kijelentkezhet a fiókjából (3. ábra).



3. ábra: Feladatválasztás

Amennyiben a tanár az jelenlét ellenőrzést választja, akkor meg kell adnia, hogy melyik osztállyal van órája és milyen tantárgyból (4. ábra).

4. ábra: Tantárgy választó

Miután kiválasztotta az adott órát a tanár, akkor megjelennek a tanórára, az osztályra és a tanulókra vonatkozó adatok. A gombra kattintva pedig elindul a jelenlét ellenőrzés, amely alapján a diák státusza megváltozik hiányzásról jelenlétre. A jelenlét ellenőrzéshez az OpenCV arcfelismerő algoritmusát használtuk. A tanulók adatai előre el lettek tárolva az adatbázisba, majd a felismerés során az eltárolt adat (kép a tanulóról) alapján beazonosítsa az algoritmus a tanulót, és a megfelelő osztályban elmenti a jelenlétet/hiányzást (5. ábra). Amennyiben nem megfelelő a fényviszony/ a tanuló távol áll a kamerától, akkor az algoritmus nem tudja felismerni a személyt. Ezen probléma megoldásán dolgozunk még.

A korábbi órák naplózási adatainál a dátum, az osztály és a tanóra kiválasztása után megjelennek a megtekinteni kívánt órák naplózási adatai (6. ábra).

A következő lépésben a nyílra kattintva megtekinthető, hogy az adott napon a tanulók közül kik voltak jelen a tanórán (7. ábra).

A munkánk széles körben használható lenne az oktatási intézményekben. Természetesen számos tovább fejlesztés vár még a

szoftverre, de a legnagyobb áttörést akkor lehetne elérni, ha a funkció a Kréta rendszerben is helyet kapna.

Magyar Hétfő 2023.02.19. 13:51 Vissza

osztály neve: osztály létszáma:

Próba Peti	Jelenlét	Hiányzás
Teszt Andi	Jelenlét	Hiányzás
Próba Pisti	Jelenlét	Hiányzás
Teszt Tamás	Jelenlét	Hiányzás
Próba Soma	Jelenlét	Hiányzás

Jelenlét 4

Ellenőrzés

5. ábra: Jelenlét ellenőrzés

← Vissza

Dátum

Tól	ig	tantárgy	osztály	Tovább
2023.03.23		Magyar	10/A	→
2023.03.24		Magyar	10/A	→
2023.03.27		Magyar	10/A	→
2023.03.29		Magyar	10/A	→
2023.03.30		Magyar	10/A	→

6. ábra: Korábbi naplózások megtekintése

← Vissza 2023.02.19. Hétfő 13:51

Magyar nyelv

osztály létszáma:
osztály neve:

Próba Peti	Jelen volt
Teszt Andi	Hiányzott
Próba Pisti	Jelen volt
Teszt Tamás	Jelen volt
Próba Soma	Jelen volt

Jelenlét 5

7. ábra: Korábbi órák megtekintése

KeySniff: Gépelt szöveg feltárása billentyűzet hangfelvételekből mély tanulással

vajdad4m

Vajda Ádám

Felkészítő tanár: Dr. Vajda Róbert

ELTE Radnóti Miklós Gimnázium, 1146 Budapest, Cházár András u. 10.

1. Bevezetés

1.1. Célkitűzés

A projekt célja egy „KeySniff” nevű innovatív megoldás kifejlesztése, amely gépi tanulási és mesterséges intelligencia technikákat alkalmaz a gépelt szöveg pontos rekonstruálására a billentyűzethangok hangfelvételeiből. A kutatás célja, hogy megoldja a gépelt szöveg rekonstruálásának kihívását a billentyűzethangokról készült hangfelvételekből, amelyek gyakran zajosak.

1.2. Téma aktualitása

A gépelt szöveg rekonstruálásának gondolata a billentyűzetről készült hangfelvételekből rendkívül fontos és jelentős a mai digitális korban. Ez a technika billentyűzeteken kívül alkalmazható ATM terminálokon, vagy esetleg gombos beléptető rendszereken.

Fontos azonban megjegyezni, hogy ehhez a technológiához jelentős biztonsági kockázat is kapcsolódik. Ha az emberek képesek a szöveget, például jelszavakat, személyes információkat vagy érzékeny adatokat csak a gépelés hallatán visszanyerni, az komoly veszélyt jelenthet a magánéletre és a biztonságra. Ezért ennek a technológiának az alkalmazhatóságát alaposan meg kell fontolni, és megfelelő biztonsági intézkedéseket kell bevezetni a felhasználói adatok védelme érdekében.

1.3. Kapcsolódó munkák

Már készültek projektek billentyűzetek akusztikus lehallgatására. Először 2004-ben tettek próbálkozást (Asonov & Agrawal, 2004) számítógépek billentyűinek, nyomógombos telefonok és ATM gombok lehallgatására. FFT-vel alakítják át a hangot a modellbe bemenő jellemzőkre. Az adathalmaz méretét nem említették. 2009-ben publikáltak egy újabb kutatást (Zhuang, Zhou, & Tygar, 2009) az akusztikus lehallgatás megvalósítására. 12 percnyi hangfelvétel kellett ahhoz, hogy a modell elérjen 90% feletti pontosságot. Egy ma fejlesztett nyílt forráskódú megoldás a Keytap3 (Gerganov, 2022). Ez mélytanulás helyett egy statisztikai modellt állít fel a lenyomott billentyűk összehasonlítására és kiválasztja a leghasonlóbbat.

WiFi jelek alapján is képesek voltak 2015-ben feltanítani egy modellt, ami képes meghatározni a lenyomott billentyűt. A publikáció (Ali, Liu, Wang, &

Shahzad, 2015) betűkként 30 mintáról számol be, ami 26 billentyű esetén 780 mintának felel meg. Ennek felvétele körülbelül 6 percet vesz igénybe. Ezen az adathalmazon a módszer 83,46% pontosságot ér el.

1.4. Kutatási terv

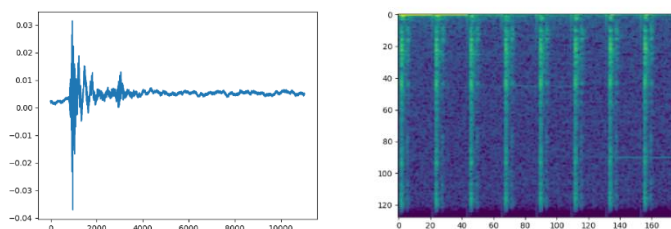
A „KeySniff” rendszer megvalósításának első lépése a billentyűzethangok hangfelvételeinek nagy adathalmazának összegyűjtése. Az adathalmaz gyűjtésén egyszerre történik a mikrofon folyamatos rögzítése, illetve a lenyomott billentyűk karakterének és a lenyomás pillanatának rögzítése. Ezután az összegyűjtött adatokon automatikusan elvégezzük a hangfelvételen a szükséges feldolgozást. Ez tartalmaz vágását, majd spektrogrammá alakítást.

A következő lépés a mély tanulási algoritmusok betanítása. A tanítási folyamat során az algoritmusok rendelkezésére bocsátjuk a spektrogram képeket és a modell kimeneti céljaként az adott pillanatban lenyomott billentyűt várjuk.

2. A kutatási folyamat

2.1. Adatgyűjtés és előfeldolgozás

A gyűjtési folyamatban egyidejűleg rögzítünk hangot egy mikrofon által a Pyaudio könyvtárral és a lenyomott billentyűket a Pygame könyvtárral. A lenyomott billentyűk mind az angol ábécé valamelyike, illetve a szóköz karakter. Így 27 lehetséges karakterből álló betűkeretet kapunk. Az adathalmaz billentyűlenyomásonként 0,5 másodperces részletek. A részletek a lenyomás pillanatától 0,1 másodperccel hamarabb kezdődnek, illetve 0,4 másodpercig tartanak. Ezeket a részleteket Mel Spektrogrammá alakítjuk. Így 1024 x 22-es felbontású képeket kapunk, amiket át tudunk alakítani 128 x 176 pixel méretű képekre.



1. ábra: Billentyűlenyomás és hozzá tartozó átméretezett spektrogram

Összesen 600 mintát gyűjtöttem, ami 5 percnyi hangfelvételnél felel meg, amik közel azonos billentyűzet-mikrofon helyzetből készültek. Egy általam készített segédprogram által tudtam figyelni a lenyomott billentyűk egyenletes elosztására

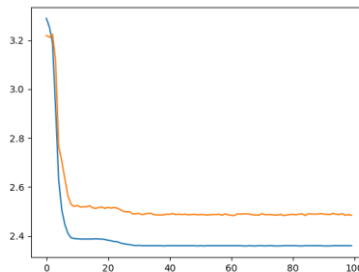
2.2. Modellezés

Az előállított modell bemenete egy 128 x 176-os méretű 1 színcsatornás spektrogram kép, a kimenete, pedig egy 27 nagyságú vektor, ami az egyes billentyűnek a valószínűsége. Mivel a modell célja, hogy valószínűségeket becsüljön a végső réteg aktiválási funkciója a softmax. A modell veszteségfüggvénye, pedig a kategorikus kereszt-entrópia. Architektúrában a VGG-16 (Simonyan & Zisserman, 2014)-ra van alapozva. Ez egy képosztályozásra készített modell szerkezet. Mivel a hangot képpé alakítottuk, kezelhetjük a feladatot képfeldolgozásként, így a konvolúciós neurális hálók (CNN-ek) alkalmazása lehetőséget biztosít a probléma megoldására.

A modell túlillesztésének elkerülése érdekében több technikát alkalmaztam. Egyik a tanító bemeneti adatok véletlenszerű torzítása, azaz eltolása és zaj hozzáadása. Másik a 0,2-es valószínűségű Dropout rétegek használata. A modell tanítása során lineáris tanulási ráta csökkentést is alkalmaztam, illetve 80-20-as tanítás-ellenőrzési eloszlást, ahhoz, hogy megfigyelhető legyen a modell teljesítménye ismeretlen adatpontokon.

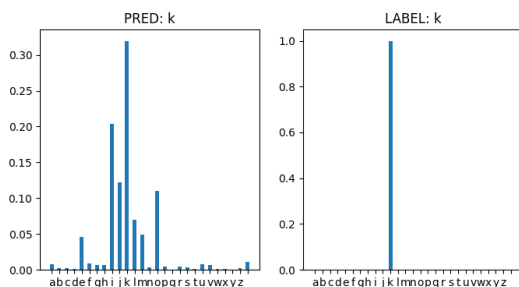
3. Eredmények

A modellt 16-os kötegmérettel tanítottam egy NVIDIA RTX 3060 videokártyán. A modell a tanulás során elkerülte a túlillesztést. Végül 100 tanulási lépés után 96,4 %-os pontosságot ért el a validációs adathalmazon.



2. ábra: A modell tanulási görbéje

A tanulási görbén megfigyelhető, hogy a modell nagyon hamar konvergál a tanítási adat globális minimumára, így a validációs veszteség azon pont után nem képes csökkenni. Ennek kiküszöbölésére lehet több adatot gyűjteni. Ez azonban szembe menne a projekt céljával, ami pedig, az, hogy kevés adaton (5 percnyi hangfelvételen) is képesek legyünk elfogadható pontossággal (>90%) megbecsülni a lenyomott billentyűket.



3. ábra: A modell egy becslése

A modellnél megfigyelhető, hogy valóban a helyet próbálja becsülni, mert a többi rosszul becsült karakter a valós cél, azaz a „k” karakter közelében helyezkednek el a billentyűzetten.

4. Jövőbeli fejlesztések

Lehetséges és ígéretes fejlesztésnek tartom a kamera alapú távolság- és mélységbecslés területén jól bevált sztereókamerás, a mi esetünkben sztereó mikrofonos adatok használatát. Ez hasonlóképpen utánozná az emberi biológiát, mint a látásnál a sztereó szenzorjaink, azaz a két szemünk. A két fül segíti az embereket a hangok irányának meghatározására, mivel képesek vagyunk észlelni a kettő érzékelés közötti eltérést, így tudunk becsülni irányt. Másik fejlesztés, pedig a címkézetlen adatokon felügyeletlen tanítás alkalmazása. Ilyenkor a modell két bemeneti spektrogramot hasonlít össze és kimenetül egy eltérési vektort ad, ami a billentyűzetten a két billentyű relatív eltérése.

5. Összefoglalás

Kutatásom célja, **hogy felhívjam a figyelmet az akusztikus lehallgatás potenciális veszélyére.** Nem kizárólag billentyűzetten, hanem más digitális eszközökön is veszélyforrást jelent. Bemutattam egy módszert, amely képes megjósolni a lenyomott billentyűket nagy pontossággal (96,4%). A megközelítésem kis mennyiségű gyűjtött adatot használ, mindössze 5 percnyi hangfelvételt. Ismertettem az általam használt módszer lehetséges jövőbeli fejlesztéseit, ami még nagyobb pontosságot eredményez.

NeuMapp

NeuMapp

Hazafi Gergő

Felkészítő tanár: Kiss Renáta

BMSZC Neumann János Informatikai Technikum, 1144 Budapest, Kerepesi út 124.

1. Bevezetés

Az iskolában sokak számára jelent igen jelentős problémát az eligazodás a többszintes, komplex felépítésű épületben. Gyakran nem tudjuk, hogy az adott terem melyik emeleten, melyik oldalon helyezkedik el. A keresett termék megtalálásán kívül további nehézséget okoz a naponta akár többször frissülő helyettesítés, az óra-, terem- és tanárrend ismerete. Az alábbi problémákra igyekeztem megoldást nyújtani a NeuMapp, Neumann iskolai térkép alkalmazással.

2. Probléma megoldásának menete

2.1. Térkép

Az elsődleges probléma, amit az alkalmazás igyekszik orvosolni, az egy olyan vizuális térkép biztosítása, ami a termék elhelyezkedése mellett képes megadott helyiséget megkeresni és azt megjeleníteni. A térkép elkészítésénél fontos szerepet játszott, hogy rezponzív és nagyítható legyen, így mindenki számára egyszerűen használhatóvá téve.

A funkció az iskolát emeletenként osztja külön szintekre (földszint, első emelet, második emelet). Az oldal tetején lévő keresőmezőbe megadhatjuk a keresett osztálytermet, vagy egyéb iskolai helyiséget, amit a program segít kiegészíteni. A térkép az 1. ábrán látható.



1. ábra: Iskolai térkép

2.2. Órarendek

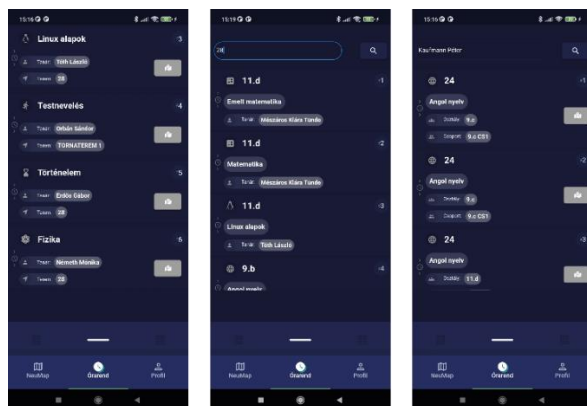
A második probléma megoldását egy személyre-szabható órarend kialakításával próbáltam teljesíteni. Az órarend használata előtt ajánlott, hogy a Profil oldalon megadjuk, hogy melyik osztályba járunk és hogy milyen csoportokba vagyunk beosztva.

Ha sikerült beállítani profilunkat, az órarendünket letölthetjük. A letöltött órarendet a program eltárolja egy szöveges fájlban így, ha nincs internetkapcsolatunk, akkor is meg tudjuk nézi, hogy mikor-hova kell mennünk órára.

Az adatok betöltése után megjelenik a jelenlegi naphoz tartozó órarendünk. Az egyes órákat megjelenítő szekciók megadják a tantárgyat, a termet, az órát adó tanárt, az óra indexét és időintervallumát. Az egyes szekcióknál található egy-egy gomb, ami megmutatja, hogy az adott terem a térképen hol helyezkedik el, ezzel a navigálást megkönnyítve.

Problémát jelenthet, hogyha valaki viszont nem az aktuális napnak az órarendjét szeretné megtekinteni, ezért lehetőség van váltani automatikus és egyénileg megadott napnak az órarendje között. A helyettesítések csak az automatikus (aktuális) nap beállításával lehet megtekinteni. A helyettesítésre egy példa a 3. ábrán tekinthető meg.

Az órarend ismerete mellett igényt tarthatunk arra, hogy tudjuk, hogy adott teremben mikor milyen órát tartanak, vagy, hogy egyes tanárokat melyik osztályteremben fognak órát tartani. E két cél kielégítésére egy-egy külön oldal felelős, ahol az oldal tetején található keresőmezőben megadhatjuk a keresett terem/tanárt, majd a program kijelzi a hozzájuk tartozó információkat. Ezt szemlélteti a 2. ábra is.



2. ábra: Balról jobbra: Órarend, Teremrend, Tanárrend

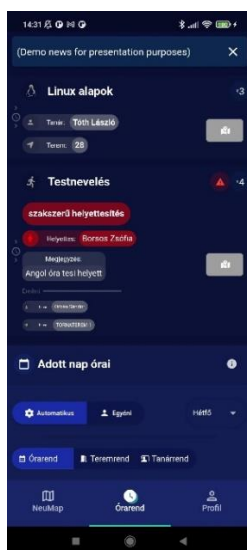
2.3. Helyettesítések

Az alkalmazás egyik legfontosabb alappillére, hogy a naponta változó helyettesítéseket képes legyen feldolgozni, és a változásokat megjeleníteni

valós-időben (real-time). A helyettesítéseket az iskola vezetősége állítja össze egy Google Sheet dokumentumba, ahonnan az applikáció képes őket lekérni, és az órarendet annak megfelelően átalakítani. Az utoljára lekért helyettesítési listát a program tárolja, amíg be nem zárjuk az alkalmazást.

Mivel a táblázat gyakran tartalmaz összevont cellákban aktuális híreket, ezért célszerűnek tartom ezek megjelenítését is, ahol őket miután elolvastuk, ki lehet törölni.

Amíg az Órarend oldalon vagyunk, 75 másodpercenként ellenőrzést végez az oldal, hogy történt-e változás a helyettesítésekben. Az alkalmazás meg nem nyitott állapotban jelenleg nem küld üzenetet, ha történt változás a helyettesítési listában. A hírekre és a helyettesítés működésére példát a 3. ábra ad.



3. ábra: Hírek és helyettesítések

2.4. Megvalósítás technikai háttere

Az applikáció Dart-ban, a Flutter keretrendszert használva íródott, számos külső plugin (package) segítségével. Az órarendi adatokat Excel fájlból olvassa be egy rövid Python program, amiket aztán feltölt egy Supabase adatbázisba, ahonnan az alkalmazás eléri.

A projektben GitHub-ot használtam a verziókövetés érdekében.

3. Elért eredmények

A lassan egy éves projekt számos hasznos feladatot képes már ellátni, amiket az 1. táblázatban láthatunk. Az alkalmazás nemcsak a diákoknak tud segítséget nyújtani a tájékozódásban és az órarendben, hanem a tanároknak is. A program ugyanazon órarendi adatokat dolgozza fel, mint amit az eKréta

rendszere, így könnyen lehet akár más iskolának is elkészíteni az órarendjét, mindaddig, amíg a helyettesítési táblázatuk felépítése megfelel a használt sablonnak. Ha az utóbbi nem teljesül, akkor azonban a program kódján szükséges a módosítás.

A térkép is egy fontos helyzet, ugyanis jelenleg a programban a saját iskolám térképe manuálisan lett beleírva, ahol minden termet egy-két függvény készít el, aminek paramétereit kézzel adtam meg. Jelenleg is folyik egy olyan verzió elkészítése, ahol a felhasználó egy Excel dokumentumot beolvasva, képes saját térképet készíteni a dokumentum adatai alapján, így akár széleskörűen elérhetővé téve az alkalmazást.

Alkalmazás funkciói oldalakra bontva	
Térkép	Vizuális térkép, automatikusan kiegészítő teremkeresés, minden emelet megjelenítése
Órarend	órarend, teremrend, tanárrend, aktuális és egyénileg megadott órarend/teremrend/tanárrend, helyettesítések, hírek/események
Profil	Felhasználónév, saját osztály megadása, hetirend, csoportok megadása

1. táblázat: Az alkalmazás funkciói

MateKIT – matematikai oktatóprogram

Fizikus módszer

Kovács János

Felkészítő tanár: Sikó Dezső, Csóti Melinda

Kecskeméti Bolyai János Gimnázium, 6000 Kecskemét, Irinyi u. 49.

4. Bevezetés

A MateKIT egy matematikai oktatóprogram, melynek fejlesztése négy éve kezdődött, miután akkori matematikatanárommal, Csóti Melindával felismertük, hogy az iskolák és a diákok már jól el vannak látva különféle digitális eszközökkel, de megfelelő szoftverekkel még kevésbé. Egy olyan alkalmazást hoztunk létre, amely ténylegesen többet nyújt annál, mint hogy megspórolja a tanárnak a táblára írást. Az alkalmazás használható tanórákon, és önálló gyakorlásra is. A program segít a differenciálásban, ami a korábbi tanítási eszközökkel nehezen megvalósítható. Az alkalmazás csökkenti a monoton tanári feladatokat, így több idő, és energia marad az órákra készülni. A tanár könnyen adhat jegyet az órai munkára, és akár dolgozatot is írthat, amelynek feladattípusaiból önállóan készülhet a diák.

5. Probléma megoldásának menete

Az iskolák Android operációs rendszerű táblagépekkel, és okostáblákkal vannak felszerelve, a diákoknak is többnyire androidos telefonjai vannak (Magyar lakosság mobileszközei 82%-án Android operációs rendszer fut). Ezért Android operációs rendszerre kezdtem el fejleszteni az oktatóprogramot, mely jelenleg 17 modulból áll. Tartalmaz 12 feladatgenerátort, melyek megoldják a diákok számára sok gyakorlást igénylő feladatok tervezését és javítását. Ezen felül tartalmaz négy játékot, és egy okos feladatgyűjteményt is. MIT App Inventorban kezdtem el a program fejlesztését, de felület korlátait felismerve váltottam Android Studioban való programozásra Kotlin nyelven. A fejlesztés során matematikatanárommal közösen meghatároztuk azokat a feladattípusokat, amelyek a legtöbb gyakorlást igénylik a diákok részéről, és a legtöbb munkát jelenti a tanároknak ezen feladatok javítása, így állt össze a 17 feladatgenerátor:

Egyenletek: 1. ábra

Megoldásukat hét típusfeladaton keresztül tanítjuk általános iskolában, a program mindegyik fajtát képes legenerálni. Egy tanárnak sok munka jól ellenőrizhető egyenleteket kitalálnia, pláne, ha törtek is szerepelnek benne. A modulban 3 nehézségi fokozat választható. Minden egyenletmegoldás egész szám.

Alapműveletek:

Lehetőség van a négy alapművelettel való számolások gyakorlására is. Több

paramétert változtathatunk, de akár előre elkészített sablonok közül is választhatunk.

Törtek:

A modul négy nehézségi fokozatot tartalmaz, de a paraméterek egyenként is állíthatóak. A diák eldöntheti, hogy szeretne-e vegyes számokat, ha igen akkor milyen tartományban. A tört részben szereplő számok ettől függetlenül variálhatóak. Megadhatjuk a kiszámítandó művelet tagszámát, vagy akár véletlenszerű hosszúságú feladatot is kérhetünk. Eldönthetjük, hogy az eredményt a program bármilyen formában, vagy csak a legegyszerűbb tört alakban fogadja el.

Római számok:

Gyakorolhatjuk az arabból római vagy rómaiból arab számokba való átváltást egy egytől négyezerig választható intervallumban.

Arányosságok:

A százalékos feladatok bevezetői. A program előre megírt szöveges feladatokhoz generál kontextusba illő számokat, így lényegében végtelen sok lehetséges példát készíthet.

Százalékszámítás:

Fontos része az általános iskolás tananyagnak, a hétköznapi életben is sokszor szükségünk van rá. Lényegében háromféle százalékos feladatot lehet készíteni, a modul mindegyik fajtát tartalmazza.

Mértékváltás:

A tanuló kiválaszthatja, hogy mely mértékegységek váltását szeretné gyakorolni, illetve maximum milyen távol legyenek azok egymástól. (pl: gramm és dekagramm közti váltást kérdező feladat sokkal gyakorlatiasabb, mint köbmilliméter és köbkilométer közötti kérdező)

Geometriai feladatgenerátor:

A 12 leggyakoribb alakzat terület, kerület és térfogatszámítási feladatait képes legenerálni. Ezen kívül az alapadatokból készíthető minden lehetséges feladatkombinációt tartalmazza. (pl. a terület ismeretében az ismeretlen oldal hosszát kérdezi)

Függvényleolvasás:

A koordinátarendszerben ábrázolt függvény képletét kell a tanulónak megadnia.

Geometria területszámítás:

Négyszéthálóra rajzol síkidomokat, melyeknek a területét kell a diákoknak kiszámolni.

A feladatgenerátorokban megjeleníthetünk egy rajztáblát, így például egy csak digitális táblával felszerelt osztályteremben van hova kidolgozni a feladatokat

A korábban megoldott feladatok megtekinthetőek. Láthatjuk a példát, hogy mit válaszolt a diák, mi lett volna a helyes megoldás, ha tévedett és a feladatmegoldás pontos dátumát is.

Az alkalmazás ezen felül tartalmaz négy játékot:

A Geometriai Vaktérkép szórakoztató módon tanítja meg a főbb síkidomokat. A játék során a program véletlenszerűen választ a 14 síkidom közül, megjeleníti a nevét, a játékos célja pedig az, hogy minél gyorsabban rákattintson. A játékos folyamatos visszajelzést kap a pontosságáról, igénybe veheti a súgót, különböző nehézségi szintekkel pedig nagyobb kihívást vállalhat. Ha a játékos egyből eltalálja a megfelelő síkidomot, akkor azt zöldre színezi a program, ha néhányat téveszt, akkor sárgára, ha pedig sokadjára sem találja meg, akkor pirosra, és kapja a következőt.

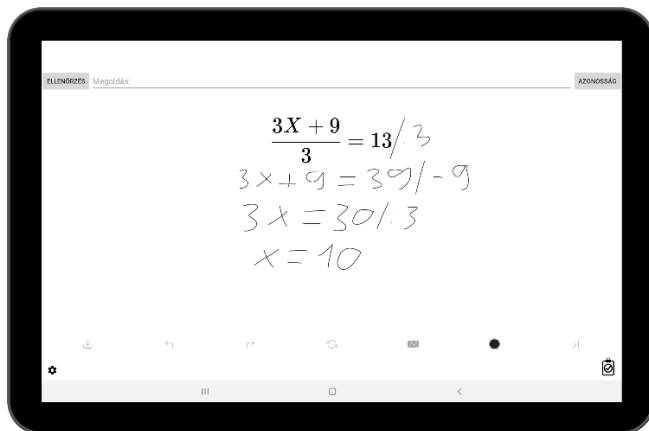
A Szögbecslés három játékot tartalmaz. Az elsőben a diáknak kell megbecsülnie egy a program által rajzolt szög nagyságát. Egy másik változatban a tanulónak kell megrajzolnia a program által kért szöget. A harmadikban a program folyamatosan rajzol egy szöget, adott foknál kell kattintania a tanulónak. Ez egy jó lehetőség a szögmérő használatának gyakorlására is.

A hatványok gyakorlására is készült egy játék. Az egyes számok hatványainak a nagyságrendjének a megismerése megalapozza a különböző számrendszerek témakörét

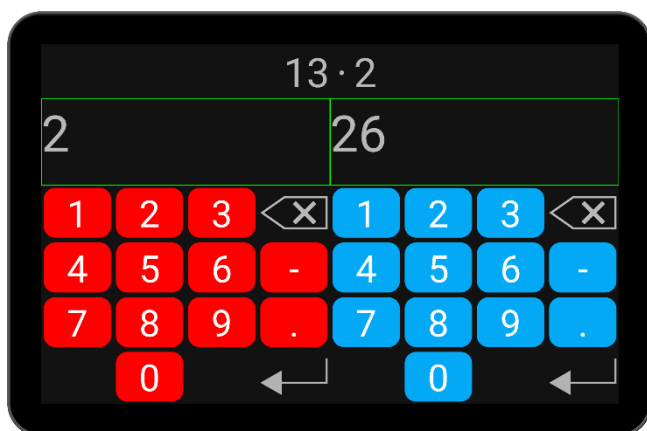
A Számkirály (2. ábra) a már évek óta sikeres pedagógiai játék okoseszközökre adaptált verziója. Lényege, hogy két tanuló egymással versenyezve old meg számolási feladatokat, aki több kört nyer, az győz. A programban különböző típusú számolásokat lehet gyakoroltatni a kezdési paraméterek megváltoztatásával. A tízes számkörben való összeadásoktól egészen fejszámoló háziversenyek lebonyolításáig mindenre alkalmas. A program törekszik értelmes, nem túlságosan nehéz feladatok készítésére. A játék végén megtekinthető, hogy az egyes körökben ki milyen válaszokat adott, mi lett volna a helyes megoldás. A felhasználói felületnek van álló és fekvő változata, így a diákok elhelyezkedhetnek egymással szemben vagy egymás mellett is.

A program része egy Okos feladatgyűjtemény: 3 témakört fed le: algebra, geometria és kombinatorika feladatokat ad véletlenszerűen a 21 feladatsorból. Az iskola értékelőrendszere egytől ötig terjedő skálára hagyatkozik, ami helyett hét szint került meghatározásra, hogy árnyaltabb képet kaphassunk a diák tudásáról. Egy tanuló szintjét a pontszáma határozza meg, amely ha megold egy feladatot növekszik, ha hibázik, akkor csökken, mint a sakkbán használt élő pontok.

5.1. Ábrák



1. ábra: Egyenlet és kidolgozott megoldása a rajztáblán (táblagép)



2. ábra: Pillanatkép a számkirályból: kék már beírta a megoldást

6. Elért eredmények

Az alkalmazás lefedi az általános iskolás matematika törzsanyagát. A program ingyenesen elérhető a Google Playből.

RepairSync

RepairSync

Magyar Virág, Laczkó Örs

Felkészítő tanár: Kovács Irén

*Budapesti III. kerületi Kerék Általános Iskola és Gimnázium,
1035 Budapest, Kerék u. 18-20.*

1. A RepairSync

A RepairSync egy olyan webalapú autószervez adminisztrációs szoftver, amely gyakorlatilag minden olyan eszközön használható, amit internethez lehet csatlakoztatni. A program megjelenése optimalizálva van az asztali számítógépektől (és laptopoktól) kezdve, a tableteken át egészen a mobiltelefonokig (400px szélességig), amely lehetővé teszi a munka különböző eszközökre történő megosztását.

A program fő funkciói:

- Ügyfél- és jármű-adminisztráció
- Munkalap-adminisztráció és munkafolyamat-menedzselés a munkalapokon
- Cikkek (járműhibák, szolgáltatások és termékek) adatbázisa
- Belső pénztár (munkalapokkal integrálva)
- Statisztika

2. Használt technológiák

A RepairSync egy teljeskörű adminisztrációs rendszer, amely számtalan egymásra épülő szinkronizált funkcióval rendelkezik, és több ember párhuzamos, kollaboratív munkáját is támogatja. Nézzük, milyen technológiák teszik ezt lehetővé.

2.1. A kliens oldal

A RepairSync kliensoldala ReactJS-ben íródott. Ez lehetővé teszi az oldal jó strukturálását, illetve a többször előforduló komponensek újrafelhasználását. A React-nek köszönhetően az oldal kezdeti betöltése után (átlagos internet sebesség mellett, meleg indítás: 700-900ms) nem kell további kattintáskor az oldalra várni, csak az adatok betöltésére (átlagos internet sebesség mellett, meleg indítás: 30ms / lekérdezés).

A RepairSync letisztult, és könnyen kezelhető admin felülete a Google Material Design-ra épülő Material-UI könyvtár alkalmazásával készült (1. ábra).

Kiemelt	Megnyitás	Név	Telefonszám	Email cím	I. Sz.	Telephely	Cím	Megjegyzés
☆	📄	Majyer Kriszta	+36 221 4232	kemjer@gmail.com	1056	Budapest	Petőfi utca 17.	Nincs
★	📄	Szűcsy Andrea	+36 302 3332	szucsand@gmail.com	6101	Miskolc	Téve utca 17.	Nincs
☆	📄	Horváth Tamás	+46 701 1820	horvath.tamas@gmail.com	1234	Kecskemét	Petőfi utca 12.	Nincs
☆	📄	Virág Antal	+36 456 6776	antalf7@gmail.com	1991	Budapest	Nádor utca 1.	Nincs
★	📄	Vilgh Béla	+36 872 2012	belagh@gmail.com	8030	Szeged	Magyar utca 3.	Nincs
☆	📄	Székely Hajnal	+36 919 2012	szekeshajnal@gmail.com	6191	Székesfehérvár	Székely utca 77.	Nincs
☆	📄	Vak Imre	+46 942 7180	vakiern@gmail.com	1080	Budapest	Róttár utca 7.	Nincs
☆	📄	Lantos László	+36 391 2012	lantoslaszlo@gmail.com	6032	Székesfehérvár	Zöld utca 7.	Nincs
☆	📄	Méhes Bertalan	+36 440 3051	malcsuk@gmail.com	6020	Zalaegerszeg	Vár utca 6.	Nincs
★	📄	Németh Sándor	+36 618 3811	nemetsandor@gmail.com	1035	Budapest	Lovász utca 18.	Nincs

1. ábra: A RepairSync ügyfél adatbázisa

2.2. A „szerver” oldal

A RepairSync Firebase-el megvalósított backend-je úgynevezett serverless server elven működik. Ennek köszönhetően a RepairSync automatikusan skálázódik, így képes kiszolgálni a hirtelen megnövekedett terhelést, úgy, hogy közben rendkívül költséghatékonyan működik. Eddigi tapasztalatok alapján körülbelül *400-450 munkaóráig* ingyenes, e fölött *0.81 ft / munkaóra*, vagy *\$0.00225 / munkaórás költség*gel lehet számolni.

3. Elért eredmények

A RepairSync fent említett összes funkciója már implementált, illetve már több mint egy hónapja folyik a szoftver éles, műhelyi tesztelése. Az említett funkciók részletesen, műhelyi igényeknek megfelelően kerültek kialakításra, több extra és kényelmi funkcióval. A szoftver további bővítéséhez számos ötletünk van, amelyek még hatékonyabbá és produktívabbá tehetik a RepairSync-et, viszont most elsődleges célunk a már meglévő funkciókat tökéletesre fejleszteni. Célunk, hogy a közeljövőben a RepairSync egy megkerülhetetlen alapkőve legyen a magyar autószerzők működésének.

Biztonságos és privát digitális kommunikáció, decentralizált hálózatokon keresztül

Rostás Raul

Rostás Raul

Felkészítő tanárok: Hegedűs Tamás Frigyes

*TSzC Széchenyi István Gazdasági és Informatikai Technikum,
2800 Tatabánya, Dózsa György út 58.*

1. Bevezetés

A digitális kommunikáció jelentősen fejlődött az elmúlt években. Ma már bárkivel azonnal kommunikálhatunk a világ bármely pontján. Sajnos egyre több, ilyen szolgáltatásokat nyújtó platformon folyik digitális megfigyelés. Elsődleges célunk egy biztonságos és privát decentralizált hálózat kifejlesztése, amely azonnali üzenetküldésre használható, és amelyet nem cenzúrázhat vagy felügyelhet semmilyen szervezet vagy kormányzat.

2. A digitális kommunikáció jelenlegi helyzete

A digitális kommunikáció jelenlegi állapota pozitív és negatív is egyben. Egyrészt azonnal kommunikálhatunk bárkivel a világ bármely pontján. Másrészt egyre több platform digitális megfigyelésnek van kitéve.

Ennek a megfigyelésnek számos formája van. Néha egy kormányzati szerv követel hozzáférést a felhasználói adatokhoz. Máskor rosszindulatú kiberbűnözők törnek be a rendszerekbe, hogy információkat lopjanak. Néha pedig maga a platform adja el a felhasználók adatait hirdetőknak, vállalati haszonszerzés céljából.

Bármilyen formában is jelentkezik, a digitális megfigyelés komoly problémát jelent. Megsérti a magánéletünket, megakadályozza, hogy szabadon kifejezzük magunkat, és akár manipulálásra is felhasználható.

3. Centralizált vs. decentralizált számítógépes hálózatok

A kommunikációs platformok nagy része teljesen központosított. Ez azt jelenti, hogy az üzeneteket vagy más média formákat a platform szerverére küldik feldolgozásra.

Nyilvánvaló, hogy Ön nem fér hozzá ehhez a szerverhez, és nem tudja, hogy mit csinálnak az adatokkal a színpalak mögött. Bár a központosított hálózatokat sokkal egyszerűbb használni, a bizalmas adatok cseréje tekintetében mindenképpen hatalmas biztonsági kockázatot jelentenek.

A decentralizált hálózatok (más néven Peer-to-Peer hálózatok) olyan számítógépes hálózatok, ahol minden felhasználó kapcsolatban áll a másik felhasználóval, akiben teljes mértékben megbízik. A decentralizált hálózatokban nincs egyetlen hibapont sem, és minden felhasználó ugyanolyan

jogosultságokkal rendelkezik, mint a többiek. A felhasználók ellenőrzését általában kriptográfiai funkciók sorozata végzi.

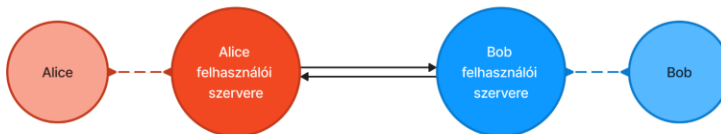
4. Probléma megoldásának menete

Mi egy decentralizált hálózat kiépítését választottuk, mivel a központosított hálózatokkal ellentétben, egy valóban privát kommunikációs platform létrehozására használható.

4.1. A hálózat felépítése

Hálózatunk 2 típusú csomópontból áll: *felhasználókból* és *felhasználói szerverekből*. (A valódi felhasználó és a *felhasználó* csomópont megkülönböztetése érdekében a csomópontok *dőlt* betűvel vannak leírva)

A *felhasználók* olyan csomópontok a hálózatban, amelyek a saját *felhasználói szerverükkel* kommunikálnak, és soha nem kommunikálnak külső *felhasználói szerverekkel*. A *felhasználói szerverek* feladata, hogy a *felhasználó* helyett külső *felhasználói szerverekre* továbbítsák a kéréseket.



1. ábra: Egy egyszerű kétszemélyes hálózat

4.2. Felhasználói szerverek közötti kommunikáció

A felhasználók hálózaton keresztül történő összekapcsolása érdekében elsődlegesen a *felhasználói szerverek* közötti kommunikáció valósul meg. A hálózatban minden egyes *felhasználói szerver* egyszerre viselkedik kliensként és kiszolgálóként, ugyanúgy, ahogyan egy központosított hálózatban is elvárható lenne a kettő viselkedése.

A fő kommunikáció a *felhasználói szerverek* között az üzenetek küldése/fogadása, a *felhasználó* üzenet történetének szinkronizálása, valamint kulcspárok lekérése egy külső *felhasználói szerver* hitelességének ellenőrzésére.

4.3. Felhasználók és felhasználói szerverek közötti kommunikáció

A felhasználó által indított műveletek esetében a *felhasználó* a saját RESTful *felhasználói szerverének* küld egy felkérést. Ezenkívül egy valós idejű RFC 6455 websocket kapcsolat jön létre a *felhasználó* és a *felhasználói szerver* között,

ahol a *felhasználói szerver* minden új eseményről vagy üzenetről publikál értesítést a *felhasználó* számára.

4.4. Az első üzenetek váltása

Példánkban 2 felhasználó létezik: *Alice* és *Bob*. *Alice* üzenetet szeretne küldeni *Bob* számára, és tudja *Bob* felhasználónevét és *felhasználói szerverének* címét: *bob@bob.hu*. *Alice* a saját felhasználói szerverén létrehoz egy szobát, és meghívja *Bob*ot. *Bob* *felhasználói szervere* értesül, hogy egy szobába került *Bob*. Lekéri a meghívó *felhasználói szervertől* a szükséges szoba információt, és hozzáadja *Bob*ot a szobához. Ezután publikálja a meghívóhoz, hogy *Bob* becsatlakozott a szobába. Ilyenkor *Alice* és *Bob* is tud üzeneteket váltani.

4.5. Egy felhasználó hitelességének ellenőrzése

A fenti példát használva, *Alice* üzenetet küld a szobába: "Helló, *Bob*!", továbbá a következő adatokat: a *feladó* (*alice.hu*), a *címzett* (*bob.hu*), a *kezdeményező* (*alice@alice.hu*), az esemény *ISO-8601 időbélyegzője*, az előző üzenet történet *hash*-je (az időbélyegzőhöz képest), és az *üzenet aláírása*, amelyet *Alice* magánkulcsával írt alá.

Amikor *Bob* *felhasználói szervere* megkapja az üzenetet, akkor:

- Meggyőződik róla, hogy a *feladó* egy érvényes *felhasználói szerver*.
- Meggyőződik róla, hogy az említett szoba lokálisan létezik, és hogy *Bob* tagja annak.
- Ellenőrzi az üzenet *aláírását* a *feladó* segítségével.
- Ellenőrzi, hogy az üzenet történetnek *hash*-je helyes-e az *időbélyeg* használatával.
- Ha a fenti feltételek mindegyike teljesül, az üzenet elküldésre kerül *Bob* részére, és *Alice* *felhasználói szervere* egy visszaigazolást kap.

4.6. Kiegészítő biztonság titkosítással

Ha *Alice* vagy *Bob* úgy dönt, hogy engedélyezik a titkosítást, akkor egy közös kulcs kiszámításra kerül a "Diffie-Hellman" kulcs megállapodásos protokoll segítségével. Ha a protokoll sikertelen, a titkosítás nem lesz engedélyezve. Ellenkező esetben a jövőbeni üzenetek a két fél által kiszámított közös kulcs használatával titkosítva lesznek. A titkosítás kikapcsolása a sikeres engedélyezés után nem lehetséges biztonsági okokból.

5. Elért eredmények

Készítettünk egy működő szerver és kliens applikációt, ahol a felhasználók biztonságos és privát módon kommunikálhatnak.

A szerver alkalmazás képes a felhasználói fiókok és kulcspárok kezelésére, üzenetek és fájlok küldésére, több eszköz között megosztott

üzenetek titkosításra és teljesen személyre szabható felhasználói profilok készítésére.

A kliens alkalmazás egy letisztult és intuitív felhasználói felületet biztosít, amit használni egyszerű és hatékony. Az üzenetek tartalmazhatnak Markdown-t. Az alkalmazás teljesen testreszabható. Saját "témák" létrehozása lehetséges CSS/SCSS használatával. Saját "bővítmények" pedig JavaScript/TypeScript használatával.

TalkHub – csapatmunka távrólól

TalkHub

Szabó Marcell

Felkészítő tanár: Dantesz Tamás

*Nyíregyházi SzC Széchenyi István Technikum és Kollégium, 4400 Nyíregyháza,
Városmajor u. 4.*

1. Bevezetés

Olyan csapatmunkát segítő alkalmazást szerettem volna létrehozni, amely vállalatok számára hasznos, saját maguk kontrollálhatják az adataikat, felhasználóikat, maguk telepíthetik a moduljait, bővíthetik, belépíthető jelenlegi infrastruktúrájukba, nem terheli őket többféle előfizetés, rugalmasan kezelhető távoli hozzáférés és ennek az egésznek egyetlen management felületen keresztül megoldható kezelése.

A self-hosted megoldások előnyei, hogy a szervezet saját maga rendelkezik adatai felett, nem kell egy külsős fél számára kiadni ezeket az információkat, így biztos lehet benne, hogy csak ő fér ezekhez hozzá. Nem kell tartani attól, hogy a harmadik felet támadás éri, ezáltal bizalmas adatokat ellopják, magunk ismerhetjük a szerverek állapotát, karbantartottságát, tudjuk, mikor voltak frissítve, milyen rendszer fut rajta.

Egyszerűen integrálható a jelenlegi vállalati rendszerbe, így amennyiben rendelkeznek VPN-nel, eldönthetik, hogy szeretnék a munkavállalóknak biztosítani az elérést, publikus eléréssel (pl <https://vallalat.hu/chat>) vagy VPN-en keresztül, helyi hálózattal.

Szever oldali telepítés előnyei, hogy a felhasználónak egyetlen appot kell telepíteni, így se külön kliens szoftvert nem kell fejleszteni, se a rendszergazdának telepíteni, backend oldalon csak telepíteni kell a kívánt modulokat és mindenki, aki ehhez a szerverhez csatlakozik, ugyan azokat a modulokat fogja látni, szerver oldalról frissíthetőek, módosíthatóak. További előnye, hogy a szervezet saját magának egységes API-nak köszönhetően fejleszthet modulokat, amelyeket telepíthet, karbantarthat.

Későbbiekben tervben van LDAP vagy Active Directory alapú bejelentkezés a beépítetten kívül, így amennyiben a szervezet rendelkezik valamelyikkel, azok felhasználói használhatóak lesznek ezen az appon belüli bejelentkezésre is, nagyban megkönnyítve a felhasználók karbantartását.

Több platform támogatása a cél, jelenleg Windows és Webes alkalmazás elérhető, de már képes működni MacOS rendszeren is, kísérleti fázisban. További cél az Android rendszer támogatása, egyszerre támogatva táblagépet és telefont is.

Tervezésnél fontos szempont volt az egy kódbasis, minden platform, valamint az operációs rendszer szolgáltatásainak minél nagyobb kihasználása, kisebb fájlméretet, natív megjelenést és jobb sebességet elérve. Támogatja a

Windows 11 újdonságait, mint a Mica style, Snap Layouts és lekerekített ablakok, Windows 10/11 WebView2 modulját, MacOS en pedig a WebView motort, valamint a transparency effektet, ottani natív megjelenés garantálása miatt.

WebRTC és WebSocket támogatása, így létrehozható videó és audio csoportos beszélgetések, valós idejű adatcsere, felesleges erőforrások használata nélkül.

HTTPS támogatása a további biztonság garantálása érdekében.

2. Probléma megoldásának menete

Kliens szerver architektúra elérése érdekében REST API-ra épül a szoftver, ez adja az egész gerincét. Backendnek MongoDB és NodeJS lett használva, saját modulokkal és saját routerrel. Lehető legfrissebb JavaScript verzióban van írva, modern szabványok támogatása érdekében. A modulok a backend oldalon vannak tárolva, a routingot is a backend oldal végzi. Amikor egy modult feltelepítenek, akkor a path eltárolásra kerül, és a kliens amikor indul, lekéri ezeket a belépési pontokat, majd betölti és hozzáadja a maga routeréhez. Minden modulnak tartalmaznia kell a nevét, modulnevet, ikonját, amit a kliensoldal megjelenít.

Jelenlegi autentikációs rendszer tárolja a felhasználó nevét, uuid-ját, hashelt jelszavát, opcionálisan az email címet. Képes a felhasználó a felhasználónevét, saját nevét, jelszavát és email címét módosítani. Felhasználónév módosítása azért megengedett, mert UUID v4 alapján történik az azonosítás, ez viszont mindenkinek egyedi.

Kapcsolati mód automatikusan felismerésre kerül, amennyiben a https-hez szükséges kulcsok megvannak, a backend automatikusan https módba vált, ha ezek pedig hiányoznak, akkor http módban működik. A kapcsolat módját a kliens képes felismerni, felhasználói beavatkozást nem igényel.

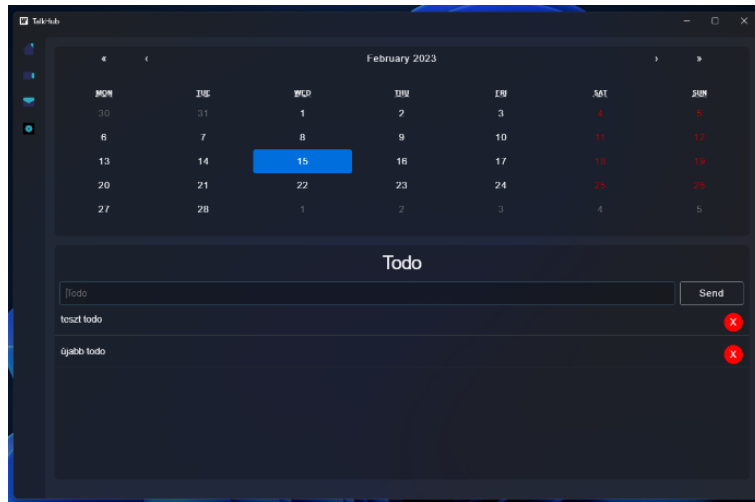
Kliens oldal Go és React felhasználásával készült, egyedi modulokkal (például az auth kliense). Windowsos és MacOS-es alkalmazás Wails (<https://wails.io>) segítségével készült, nem pedig Electronnal, így lényegesen kisebb alkalmazás készül, és az operációs rendszer beépített eszközeit használja, nem pedig beépített Chromiumot, valamint használható a natív design elemek is, így ezt jobb választásnak tartottam.

Authentikációs modul kliensoldali változata jelenleg újrírás alatt, így csak egy alapszintű azonosításra használható modul van ideiglenesen benne, amíg a másik el nem készül.

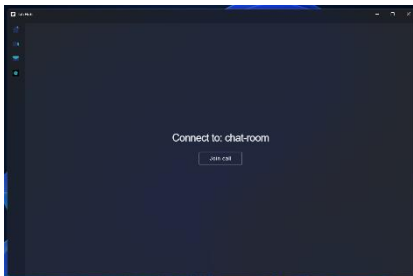
3. Elért eredmények

Windowsos appot mutatom be, jelenlegi formájában.

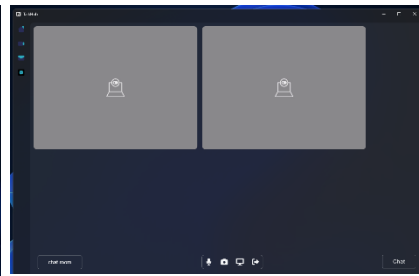
Ebben a telepítésben a kezdőoldal egy feladatütemező található (1. ábra), egy naptárral és egy todo-val. A feladatokat az egész csoport látja, fejlesztés alatt áll, hogy a naptáron belül lehessen todo-kat megjeleníteni, létrehozni, szerkeszteni, határidők egyszerűbb követésére.



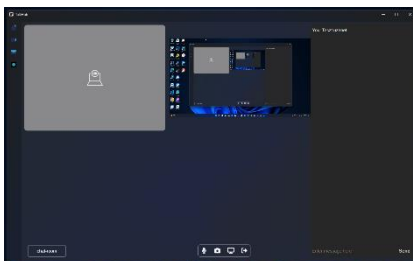
1. ábra: A feladatütemező



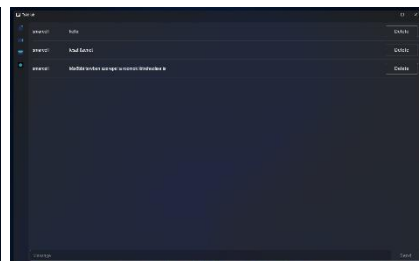
a) Csatlakozás



b) Eszközök engedélyezése



c) Képernyőmegosztás



d) Üzenetek

2. ábra: Chatszoba

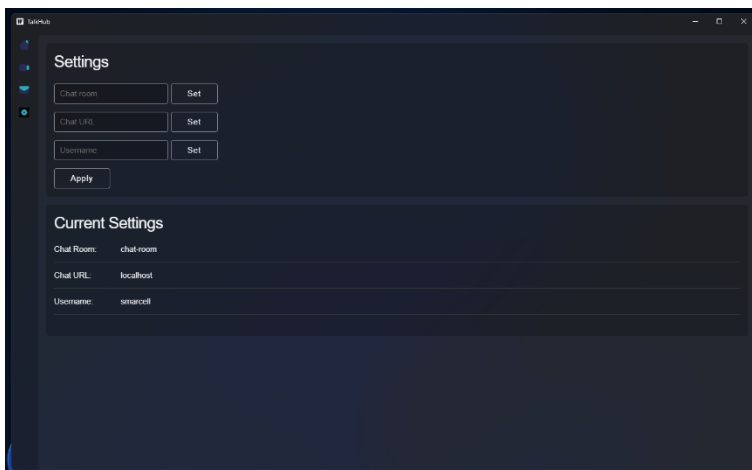
A 2a. ábrán látható oldal a webes meetinghez való csatlakozást hivatott szolgálni, a beállított chatszoba nevének megjelenítésével. Minden felhasználó képes tiltani, engedélyezni a mikrofonját, kameráját, valamint képernyőt osztani, chatelni közben, ezzel segítve a prezentációk tartását is (2b. és 2c. ábra).

A 2d. ábrán látható a chat jelenlegi formája, folyamatban vannak chatszobák implementálása, így a beszélgetés jól elkülöníthető lesz, valamint a válaszolás lehetősége is.

Ez pedig a beállítások. Ez egy fix komponens, nem választható, mivel a kapcsolati beállításokat itt lehet megadni, valamint ha az újraírt hitelesítő modul felhasználói adatait is itt lehet majd lekérni. A „chat room” opció át fog kerülni a videóhívás részre, ahol kiválaszthatja az engedélyezett szobákat, így ezt nem kell manuálisan állítani majd, és nem függ majd az alkalmazás beállításaitól. Megtekinthetők a beállítások is alul, tájékoztatási célokból.

Miben más mint a többi alkalmazás? Igény szerint bővíthető, nem függ külső szerverektől, egyszerű karbantarthatóság és központi vezérlés a fő erényei.

Az alkalmazás azoknak lehet hasznos, akik nem szeretnék drága előfizetéseket fizetni, de szükségük van vállalati kommunikációra, csapatmunkát elősegítő eszközöket használni, például Code hosting (git), boardok (kanban), issue tracker. Ezek a modulok a közeljövőben le lesznek implementálva, és az alapértelmezett csomagban telepíthetőek is lesznek. További modulokat készíthet a szervezet, így amennyiben egyedi igények vannak, azt is egyszerűen, ugyan ahhoz a platformhoz elkészíthetik.



3. ábra: Beállítások

LEX

chmod 777

Márton Gábor

Felkészítő tanár: Lestál Máté

*Fáy András Görögkatolikus Technikum és Szakgimnázium,
3529 Miskolc, Jászi Oszkár u. 1.*

1. Bevezetés

A LEX (LazerEX) egy könnyű, nyílt forráskódú kezelőfelület a reddit-hez, ami Go (golang)-ban, és az SASS CSS keretrendszer segítségével lett készítve.

A probléma a reddit-tel az, hogy nagyon lassú, és sok internetet használ fel. Ezeken kívül a felhasználói felülete lefagyhat ok nélkül, akár több másodpercre is (akár 20mp-ig is, ha régi a számítógép). Ezeket a problémákat bármely böngészőn, amin próbáltam (Chrome, Firefox, Brave, stb.) lehetett tapasztalni. A Lex, a reddit-tel szembe 60%-kal kevesebb internetet használhat fel, miközben nagyon hasonló kép-/videóminőséget biztosít. Ezen kívül, az interaktívvá válás ideje átlagosan körülbelül 800ms-1,2s (az összes kép, szkript, stíluslap betöltésével együtt). A Lex felhasználói felület kinézetének célja az, hogy hasonlítson a reddit-hez, miközben pár elemet kompaktabbá tesz.

A LEX forráskódja a <https://github.com/cmd777/lex> címen található.

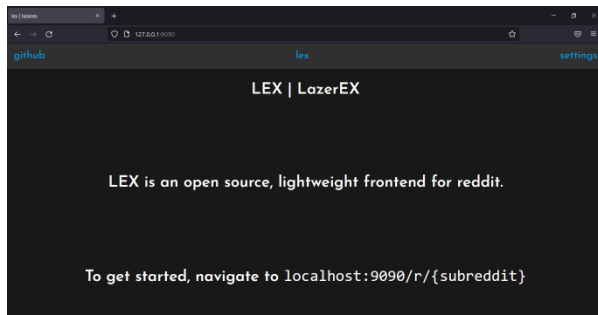
A forráskód mellett a repository tartalmaz néhány hasznos README-t, amelyek útmutatást tartalmaznak a projekt építéséhez/használatához.

2. Probléma megoldásának menete

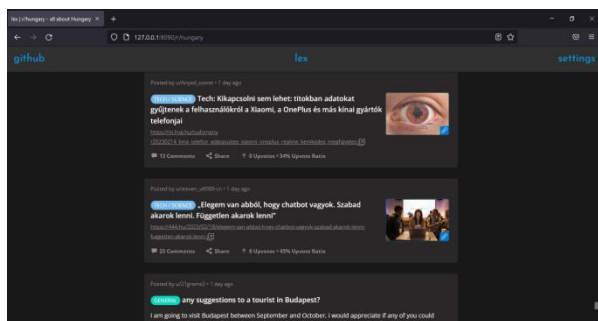
A cél az volt, hogy csináljak egy olyan reddit felületet, amihez nem kell fiók, nem kell különböző böngésző beállításokat változtatni, és az, hogy JavaScript nélkül is működjön. Szerencsére, a reddit biztosít külön JSON végpontokat, amelyek segítségével elég könnyen lehetett megvalósítani, de ez semmiképpen sem azt jelenti, hogy tökéletesen működik. Például: A reddit nem biztosít nyers videó fájlokat, csak DASH / HLS videókat. A DASH videó jól működik, csak egy problémája van: Nincsen hang. A HLS videóval nincs ilyen probléma, de nem lehet HTML <video>-ba rakni, és szükség van egy JavaScript könyvtárra (mint például a hls.js-re) hogy lejátszjuk. A JavaScriptet még azért használok, mert lehetővé teszi azt, hogy ne keljen új oldalra menni, hogy betöltsünk új posztokat.

Jelenleg, a JavaScriptet nem lehet kikapcsolni, és szükséges az oldal működéséhez, viszont be lesz építve egy konfiguráció szekció, ahol majd ki lehet kapcsolni.

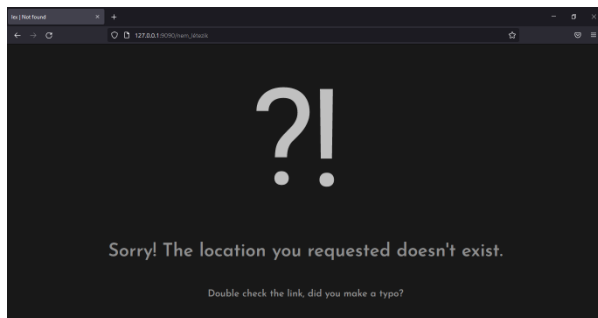
2.1. Ábrák



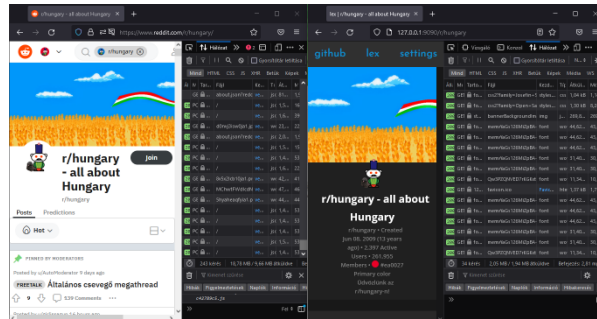
1. ábra: A LEX főoldala



2. ábra: A külső hivatkozásokhoz indexképek jelennek meg



3. ábra: Egy külön 404-es oldalt mutatja, ha az elérési út helye nem létezik



4. ábra: Az internethasználat összehasonlítása (a baloldalon a Reddit, a jobb oldalon a LEX.)

3. Elért eredmények

Ahogy a 4. ábrán látható, a LEX akár 9-szer kevesebb internetet fogad, és 5-ször kevesebb adatot küld, és a betöltés szempontjából sokkal gyorsabb.

Ez azért jó, mert a gyengébb internet-hozzáféréssel rendelkező eszközök gyorsabban tudják betölteni a LEX-et, mint a Reddit-et.

Együttműködő partnereink

Együttműködő partnereink



ExxonMobil Hungary Kft.
www.exxonmobil.hu



3i Fejlesztő és Szolgáltató Kft.
www.3i.hu



EPAM Systems Kft.
www.epam.com

Morgan Stanley

**Morgan Stanley Magyarország
Elemző Kft.**
www.morganstanley.com

s:)nrisa

Sonrisa Technologies
www.sonrisa.hu



Digital Kft.
www.digital.co.hu



TEConcept Hungary Kft.
www.teconcept.hu



Optin Kft.
www.optin.hu



Creativ_IT
www.creativit.hu



NOKIA
www.nokia.com



KNORR-BREMSE

**Knorr-Bremse Vasúti Jármű
 Rendszerek Budapest**
www.rail.knorr-bremse.com/en/hu/



bitotron
 We understand We deliver

Bitotron Kft.
www.bitotron.com



Lufthansa Systems

**Lufthansa Systems Hungária
 Kft.**
www.lhsystems.com



Accenture
www.accenture.com



CAS Software Kft.
www.cas-software.hu



**SZTE TTIK Informatikai
 Intézet**
www.inf.u-szeged.hu



**SZTE Természettudományi és
 Informatikai Kar**
www.sci.u-szeged.hu

Jegyzetek



Kiadta: SZTE TTIK Informatikai Intézet

Készítette: Dr. Németh Gábor

Design: Dr. Németh Gábor

Nyomda: Innovariant Nyomdaipari Kft.

Készült: 100 példányban

Együttműködő partnereink:



SZTE TTIK
INFORMATIKAI INTÉZET



s:)nrisa

ExxonMobil

<epam>



KNORR-BREMSE

NOKIA

Morgan Stanley

<reativ_it



TEConcept
Hungary Kft.



bitotron
➤

OPTIN Digital

accenture



Lufthansa Systems