

UniLED Display Kijelző Vezérlő Prototípusának Záró Fejlesztési Jegyzőkönyve

Műszaki Záróbeszámoló és Disszeminációs Dokumentáció (VEKOP-2.1.7-15-2016-00568)



Alkotni szeretünk!

Megnevezés	Érték
Projekt azonosító	VEKOP-2.1.7-15-2016-00568
Támogatási szerződés száma	IKT-2018-600-I1-00005852 IKT-2019-600-I1-00002010/0000098
Projekt címe	Moduláris táblavezérlő prototípusának kifejlesztése és piaci bevezetése
Termék megnevezése	UniLED Display Kijelző Vezérlő Prototípus (HW: 274.1.0 BV11, FW: CS143.34)
Jelentés verziója	1.0 (Prototípus Záró Jelentés)
Kibocsátás dátuma	2025.08.28.

Tartalom

1.	A piaci igény diagnóza és a hagyományos vezérlők korlátainak elemzése.....	3
2	A vezérlő rendszer koncepciójának születése	3
3	Projekt rövid bemutatása és alap koncepció	4
3.1	Projekt összefoglalás és alapvető hardver architektúra:	4
3.2	Több szintű, egységes kommunikációs interfész:	5
3.3	Integrált szenzoros ökoszisztéma és riasztás kezelés:	5
4	Fejlesztési prioritások és funkcionális részletezés.....	5
4.1	Megbízható rendszer- és adatintegritás:	6
4.2	Fail-safe firmware frissítés:	6
5	Dinamikus kijelző- és tartalommenedzsment	6
5.1	Field-beállítás és típusok:	6
5.2	Riasztási és menedzsment mechanizmusok:	7
6	Fejlesztésben részt vevők.....	8
6.1	A fejlesztőcsapat struktúrája, szerepkörök és feladatok:.....	8
7	Projekt fejlesztésének menetének leírása (tesztelések, eredmény).....	9
7.1	Fejlesztési idővonal:	9
7.2	Tesztelési megközelítés és eredmények:	9
8	Konklúzió	10

1. A piaci igény diagnózisa és a hagyományos vezérlők korlátainak elemzése

A UniLED Display projekt alapvető problémája az volt, hogy az ipari és közterületi kommunikációs rendszerekben alkalmazott kijelzővezérlő megoldások nem voltak képesek megfelelni a modern, hibrid kommunikációs elvárásoknak és az Ipar 4.0 által megkövetelt robusztus távmenedzsmentnek. A fejlesztés megkezdésekor a piacon elérhető vezérlő egységek általában csak egyetlen kommunikációs csatornát támogattak, mint például a lokális RS485 buszt vagy az egyszerű, vezetékes LAN kapcsolatot, ami jelentősen korlátozta a telepítési helyszínek választását és a működtetési rugalmasságot.

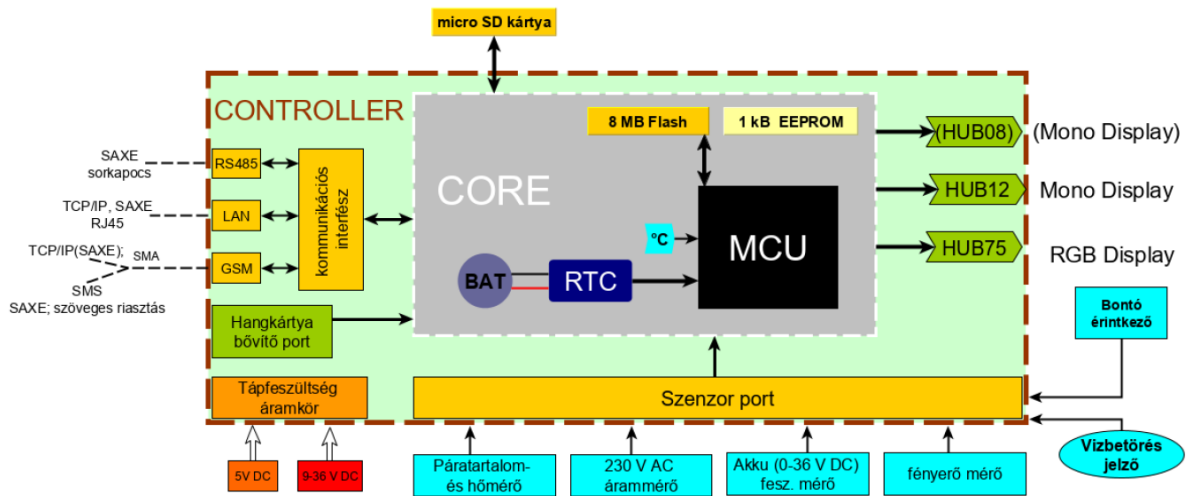
Ez a kommunikációs unilateralitás komoly üzemeltetési nehézségeket okozott ott, ahol a vezetékes infrastruktúra kiépítése költséges vagy lehetetlen volt, így a mobilhálózaton keresztüli stabil, kétirányú kommunikáció hiánya kritikus volt. Emellett a hagyományos vezérlők komoly hiányosságokat mutattak az adatintegritás és a fizikai biztonság terén. Kritikus folyamatok, mint például a firmware frissítése, könnyen megszakadhattak zajos vonalakon (RS485) vagy instabil mobilkapcsolaton (GPRS) keresztül, ami a készülék tartós működésképtelenségét okozhatta. Végül, a karbantartás szempontjából elengedhetetlen volt a helyi környezeti állapotok (például belső hőmérséklet, feszültség szint, fizikai behatolás) monitorozásának hiánya, ami megakadályozta a proaktív hibaelhárítást és a diagnosztikát.

2 A vezérlő rendszer koncepciójának születése

A fenti korlátok elhárítására született meg az UniLED Display koncepciója. A cél egy olyan univerzális vezérlő platform létrehozása volt, amely képes a különféle kommunikációs csatornákat egységes protokollon keresztül kezelni, miközben maximális hardveres és szoftveres üzembiztonságot nyújt. A vezérlő architektúrája egy nagy teljesítményű ARM Cortex M4 alapra épült, ami lehetővé tette a Field-orientált logikával történő, nagy sűrűségű adatfeldolgozást. A stratégiai cél az volt, hogy a rendszer ne csupán a kijelző tartalmát vezérelje, hanem komplett távfelügyeleti végpontként is funkcionáljon. Ennek érdekében a koncepció kulcsfontosságú elemei a redundáns biztonsági mechanizmusok lettek: a kommunikációs protokoll csomagjait kettős ellenőrző összeggel láttuk el, és egy többszintű, megszakítás esetén is helyreállítható (fail-safe) firmware frissítési mechanizmust dolgoztunk ki. Ez a megközelítés garantálta, hogy az eszköz még a legkritikusabb műveletek során is megőrizze működőképességét.

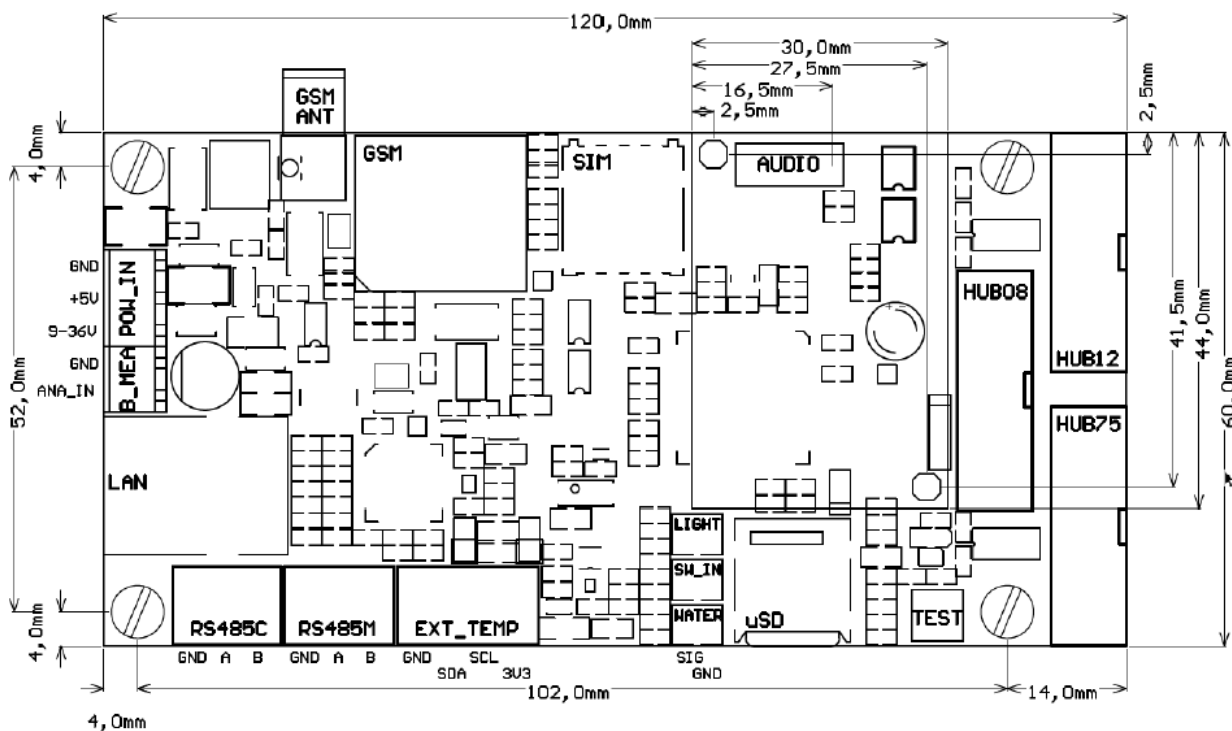
3 Projekt rövid bemutatása és alap koncepció

3.1 Projekt összefoglalás és alapvető hardver architektúra:



1. Ábra: Az architektúra szemléltetése

Az UniLED Display kijelző vezérlő prototípus egy beágyazott rendszer, amely magja egy ARM Cortex M4 típusú mikrovezérlő. A memóriakezelés a feladatokhoz optimalizált. A képtároláshoz és a fontok átmeneti tárolásához 8 MB FLASH memória áll rendelkezésre, míg a konfigurációs paraméterek tárolására 1 kB EEPROM-ot használnak. A jövőbeli hangfájl-lejátszási funkciókhoz egy microSD kártya foglalát került kialakításra. A kijelzők csatlakoztatására HUB12 (Monokróm) és HUB75 (RGB) csatlakozók állnak rendelkezésre, amelyek maximum 192 x 64 pixeles kijelzőméret kezelését teszi lehetővé. A pontos időszámítást egy Real Time Clock (RTC) chip biztosítja, melyet egy 3 V, 5 mAh akkumulátor táplál, garantálva az időszámítás folyamatosságát áramkimaradás esetén is.



3.2 Többszintű, egységes kommunikációs interfész:

A vezérlőrendszer egyik meghatározó jellemzője, hogy a kommunikáció Master-Slave jellegű, ami azt jelenti, hogy az eszköz (Slave) kizárólag a Master parancsait hajtja végre, és azokra válaszol. Az alkalmazott protokoll minden támogatott fizikai csatornán egységes. A kommunikációs csatornák sokfélesége biztosítja a telepítési rugalmasságot. A vezetékes ipari kommunikációt az RS485 sorkapocs biztosítja, amely ESD védelemmel van ellátva, és minimum 1 Mbps sebességet támogat, alapértelmezett beállításokkal: 115200 BAUD, 8 adatbit, 1 stopbit, nincs paritás és nincs handshake. A hálózati kommunikáció két fő formában lehetséges. A LAN (RJ45 foglalat) TCP/IP kommunikációt biztosít, alapértelmezett TCP Szerver módban, 10001-es porton. A mobilhálózati kapcsolatot GSM/GPRS modul kezeli (900, 1800, 1900 MHz támogatással, SIM kártyával), amely TCP Kliens módban csatlakozik egy szerverhez.

A kommunikációs csatornák és alapbeállítások összefoglalása a következő táblázatban látható:

Csatorna	Protokoll	Alapvető paraméterek	Sebesség/Mód
RS485	Soros	115200 BAUD, 8N1	Min. 1 Mbps (ESD védelem)
LAN (TCP/IP)	TCP Szerver	IP: 192.168.1.100, Port: 10001	TCP Szerver mód
GSM/GPRS	TCP/IP, SMS	IP: 92.249.219.44, Port: 17199	TCP Kliens mód

3.3 Integrált szenzoros ökoszisztéma és riasztás kezelés:

A vezérlő nem csupán a kijelző tartalmának megjelenítéséért felelős, hanem egy kiterjedt szenzoros ökoszisztémát is kezel az eszköz fizikai integritásának és környezetének felügyeletére. Ez a képesség teszi az eszközt valós idejű távfelügyeleti végponttá.

A beépített szenzorok közé tartozik a dobozon belüli hőmérő (-40°C -tól 125°C -ig mér, $0,1^{\circ}\text{C}$ pontossággal), amely a túlmelegedés detektálására szolgál. A fizikai behatolás elleni védelmet egy szerelhető bontó kapcsoló (Dobozbontás érzékelés) és egy vízbetörést érzékelő NYÁK biztosítja. A környezeti paraméterek mérésére szolgál egy külsőleg elhelyezhető páratartalom- és hőmérő. Ezen felül az eszköz méri az akkumulátor (vagy annak hiánya esetén bemeneti tápfeszültséget $0-36\text{ V DC}$) feszültségét, és képes kezelni egy külső 230 V AC árammérőt (OR-WE-514) az RS485M vonalon keresztül. A külső fényerősségmérő (LIGHT csatlakozó) a környezeti fényviszonyok alapján szabályozza a kijelző fényerejét.

Az ezen szenzorok által gyűjtött kritikus állapotok az Alert (Vészjelzés) és Error (Rendszerhiba) mezőkben összegződnek, amelyek a 11h/E0h/E2h státuszlekérdező parancsok válaszaiban kerülnek továbbításra. Az Alert mező bitjei magukban foglalják a dobozbontás, a vízbetörés, a belső hőmérséklet túlmelegedés és az alacsony feszültség érzékelését.

4 Fejlesztési prioritások és funkcionális részletezés

A prototípus fejlesztése során a legmagasabb prioritást a rendszer megbízhatósága, az adatintegritás és a kijelződinamika kapta, tükrözve a kritikus ipari felhasználásra szánt eszköz elvárásait.

4.1 Megbízható rendszer- és adatintegritás:

A kommunikáció megbízhatóságának biztosítása érdekében a protokoll minden parancsa és válasza egy keretbe van foglalva, amely tartalmaz egy kétszintű ellenőrző mechanizmust. A csomag kezdetét az STX (02h) és a végét az ETX (03h) jelöli. Az adatintegritás ellenőrzésére a CSSZ1 és a CSSZ2 szolgálnak. A CSSZ1 egy egyszerű aritmetikai összeg, melyet modulo 256-tal normalizálnak: $CSSZ1 = (adat_0 + adat_1 + \dots + adat_n) \bmod 256$. A CSSZ2 ezzel szemben egy kizáró vagy (XOR) művelettel számított ellenőrző szám: $CSSZ2 = adat_0 \text{ XOR } adat_1 \text{ XOR } \dots \text{ XOR } adat_n$. Ez a kettős integritásellenőrzés alapvető fontosságú. A redundáns számítás célja, hogy minimalizálja az adatsérülés kockázatát, különösen azokon a zajos fizikai vonalakon (pl. RS485) vagy a csomagvesztésre hajlamos vezeték nélküli GPRS csatornán, ahol egyetlen checksum nem feltétlenül elegendő a beágyazott rendszer kritikus adatainak védelméhez.

4.2 Fail-safe firmware frissítés:

A távoli firmware frissítés (SW frissítés) az eszköz legkritikusabb művelete, amelynek során a rendszer rendelkezik a működőképességének helyreállítását lehetővé tevő mechanizmusokkal. A frissítés három fázisban zajlik, biztosítva a magas üzembiztonságot.

1. **Előkészítés (06h):** Először a 06h paranccsal törlik az átmeneti flash memóriát, ami a későbbi programkód fogadására szolgál.
2. **Adatírás (0Ah):** Ezt követően a firmware fájlt maximálisan 996 bájtos csomagokra bontva töltik le az átmeneti memóriába a 0Ah paranccsal. Minden sikeres írási csomagra a vezérlő visszajelzést küld (0Bh), biztosítva a Master számára, hogy az adatok a célmemóriába kerültek.
3. **Végrehajtás (0Ch):** A teljes firmware letöltése után kiadott 0Ch parancs indítja el a tényleges program memória írását. Erre a parancsra, és az utána következő belső műveletekre a processzor már nem válaszol, hanem megkezd a belső ellenőrzéseket és a másolást.

A processzor minden újraindítás után ellenőrzi a programmemóriát és az átmeneti flash tartalmát. Amennyiben hiba lép fel a frissítési folyamat alatt (például tápfeszültség szünet), a rendszer megpróbálja kijavítani a hibát, vagy visszaáll a korábbi, működőképes firmware-re. Ezzel a mechanizmussal kizárjuk annak lehetőségét, hogy a hardver működésképtelenné váljon egy megszakított frissítés miatt.

5 Dinamikus kijelző- és tartalommenedzsment

5.1 Field-beállítás és típusok:

A vezérlő 80 független Field (mező) kezelésére képes, 0-tól 79-es sorszámmal. A Field-alapú architektúra rugalmasan oszthatja fel a kijelző területét. A Field-ek beállítása a 60h paranccsal történik, ahol megadják a bal felső X és Y koordinátát, a szélességet és magasságot pixelben, a betűtípust, a színt, a háttérszínt és az igazítást.

A Field-ek dinamikus megjelenítésre képesek az alábbi típusokban:

1. Óra (másodpercenként frissül, a kettőspont ":" villog, ha van online kapcsolat)
2. Hőmérséklet (külső szenzor értéke)
3. Hőmérséklet és páratartalom
4. Páratartalom (külső szenzor értéke)
5. String (szöveg)

A V2.3-as verzió bevezetett egy kritikus bővítést a stringek kezelésében. Az első három Field (0, 1, 2) képes akár 1000 karakter hosszúságú szöveges tartalom fogadására és megjelenítésére is, amennyiben a String effekt beállításánál Scroll vagy Page effektust alkalmaznak. Normál, statikus megjelenítés esetén a karakterlimit 128 karakter. A Field tartalmát a 68h paranccsal lehet írni, amely lehetővé teszi a Flash, Inverz, 3 pont (C), Page (P), és Scroll (S) effektek konfigurálását.

A Field-ek definíciójánál nincs automatikus ellenőrzés vagy védelem az átlapolódás ellen; a felhasználóra van bízva a helyes megjelenítés kialakítása. Azonban a megjelenítés sorrendjét szigorú prioritási logika határozza meg, amelyet a Field aktiválási parancs (80h) kiadásának időrendje szabályoz. Mindig a legutoljára kiadott aktiválási parancs Field-je kerül felülre, azaz lesz a legerősebb, míg a korábban aktivált Field-ek gyengébbek és az alattuk lévő tartalom jelenhet meg. Ez a prioritási rendszerezés hatékony módszer a renderelési sebesség maximalizálására, elkerülve az összetett szoftveres konfliktuskezelési algoritmusokat, amelyeket a hagyományos grafikus motorok igényelnek.

5.2 Riasztási és menedzsment mechanizmusok:

A vezérlő autonóm távmenedzsment képessége a proaktív állapotfelügyeleten és az aszinkron riasztásküldésen alapul.

A rendszer a státusz lekérdezésekor (10h) két kategóriába sorolja a detektált problémákat:

2. **Alert mező (Vészjelzés):** Érzékeny, fizikai események, amelyek SMS-ben riasztást generálnak.

- Dobozbontás érzékelés (0. bit)
- Vízbetörés érzékelés (1. bit)
- Belső hőmérséklet túlmelegedés (2. bit)
- Feszültség alacsony (3. bit)

3. **Error mező (Rendszerhiba):** Hardveres és szenzorhibák.

- Belső óra hibás beállítás (0. bit)
- Külső EEPROM/FLASH hiba
- Szenzorhibák (belső hőmérő, külső hőmérő + páratartalom, külső légnyomás, 230V AC fogyasztásmérő).

Riasztási flag változásakor (mind az igaz, mind a hamis állapot elérésekor) az eszköz SMS-t küld a GSM Admin telefonszámra, ami a paraméterlistában állítható. A billegő értékek miatti túl sok SMS kiküldésének elkerülése végett egy paraméterrel beállítható a riasztások közötti minimális türelmi idő.

A protokoll V2.2-es verziójában bevezetett E0h és E2h parancsok kritikus szerepet játszanak a hálózati sávszélesség és a reakcióidő optimalizálásában. Az E0h parancs az automatikus státuszküldési funkcióhoz kapcsolódik, amely paraméterrel állítható periódusidővel (37. Aszinkron automatikus státusz parancs küldési periódus idő, alapértelmezett 3 másodperc) küldi el az eszköz állapotát anélkül, hogy a Master lekérdezné. Az E2h parancs, amelynek tartalma megegyezik a 10h/11h státusszal, csak a riasztási események (Alert flag változás) bekövetkeztekor kerül kiküldésre (ha engedélyezve van), így biztosítva az azonnali tájékoztatást a kritikus hibákról. A működést nagy mértékben befolyásoló paraméterek közül kiemelték a riasztási küszöbértékek és a kommunikációs viselkedést szabályozó beállítások.

No	Megnevezés	Típus	Alapérték	
4	Belső hőmérséklet riasztási küszöbszintje	s08	70	Biztonsági limit. Túllépés esetén Alert SMS küldése.
5	Szünetmentes feszültség riasztási küszöbszintje	u08	0	Akkuszint felügyelete. Alacsony feszültség riasztás.
11	GPRS Periódus idő	u16	10 perc	Csatlakozási gyakoriság Offline állapotban.
23	GPRS Online állapot	u08	1	1: Folyamatos kapcsolat tartása a szerverrel (Client mód).
37	Aszinkron automatikus státusz parancs küldési periódus idő	u16	3 sec	E0h parancs küldési gyakorisága (0: kikapcsolva).

6 Fejlesztésben részt vevők

A UniLED Display kijelző vezérlő prototípusának komplexitása megkövetelte, hogy a fejlesztés multidiszciplináris szakértelemmel történjen, amely magában foglalja a hardver, a beágyazott szoftver és a kommunikációs protokoll szakértőit.

6.1 A fejlesztőcsapat struktúrája, szerepkörök és feladatok:

A projekt sikeréhez elengedhetetlen volt a hardvertervező, a firmware fejlesztő és a minőségbiztosítási szakemberek szoros együttműködése. A hardver és elektronikai mérnök feladata volt az ARM Cortex M4 alapú NYÁK (274.1.0 BV11) tervezése, biztosítva a megfelelő perifériás integrációt (pl. memória, kommunikációs csatornák, ESD védelem). A projektmenedzser felelt a projekt egészének irányításáért, a határidők és a pályázati elvárások teljesítéséért.

A csapat legfontosabb technikai szereplője a fő beágyazott szoftver fejlesztő volt, aki a teljes firmware és a kommunikációs protokoll (V1.0-V2.4) megalkotásáért, dokumentálásáért és folyamatos

frissítéséért felelt. A tesztmérnökök felelőssége volt a rendszer robusztusságának validálása, különös tekintettel a Field-vezérlési logika és a fail-safe firmware frissítési eljárások megbízhatóságára.

7 Projekt fejlesztésének menetének leírása (tesztelések, eredmény)

7.1 Fejlesztési idővonal:

Az alábbi lista tartalmazza, hogy a dokkolók és az oszlopok egyes alkatrészeinek milyen EU irányelveknek kell megfelelniük.

A UniLED prototípus szoftveres fejlesztése 2019 májusában vette kezdetét a protokolltervezet (V1.0) elkészítésével, majd 2020 februárjára érte el a stabil V2.4 verziót. Ez az iteratív fejlesztési folyamat két fő fázisra bontható.

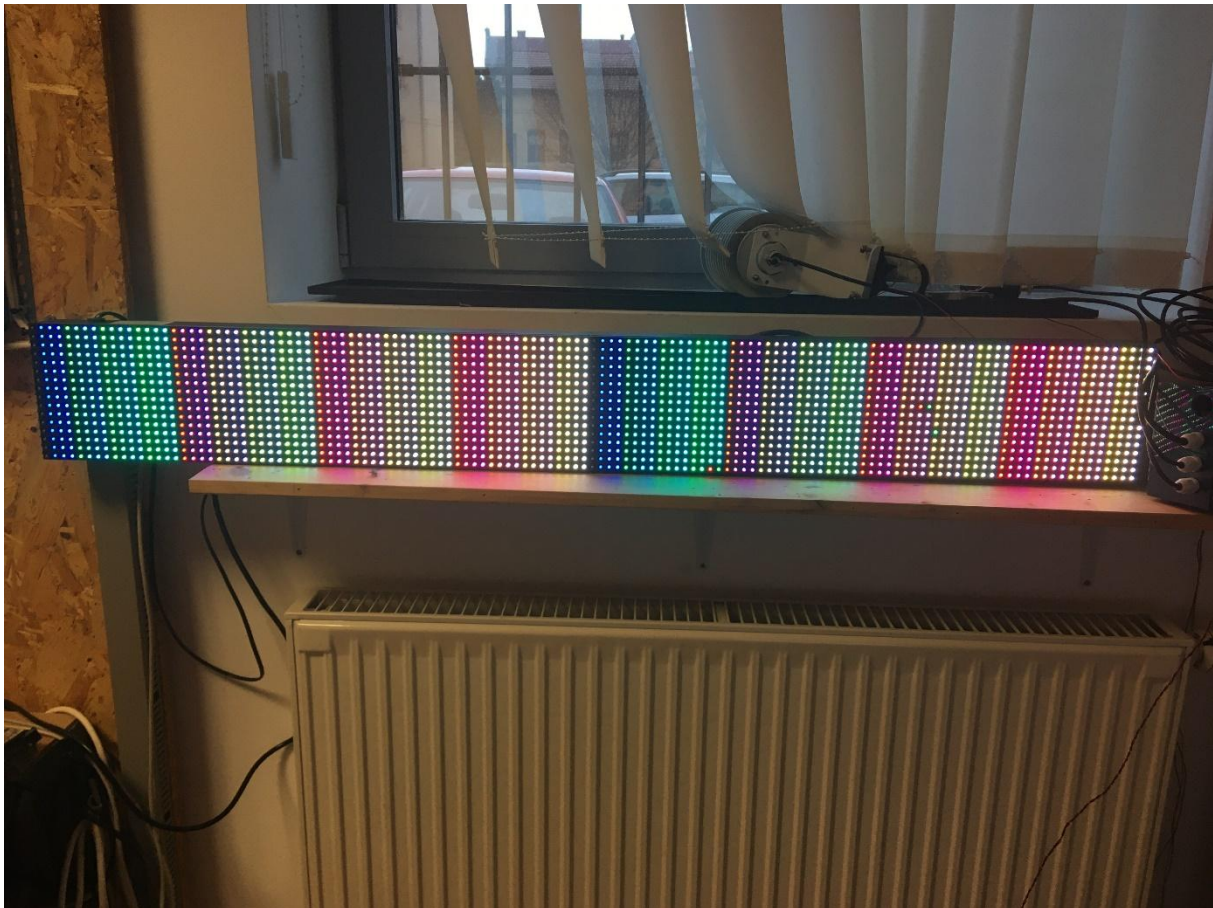
Az első fázis (2019. május – szeptember, V1.0-V2.0) a hardveres integrációra és az alapvető Field funkciók megvalósítására fókuszált, elérve a „Megvalósult eszköz protokoll leírása” állapotot (V2.0). A második fázis (2019. október – 2020. február, V2.1-V2.4) a robusztusságra és a funkcionális skálázhatóságra koncentrált. Ekkor történt meg a címzési tartomány pontosítása (V2.1), az aszinkron állapotjelentő parancsok (E0h, E2h) bevezetése a proaktív menedzsment érdekében (V2.2), valamint a Field-szám és a string hosszának növelése (V2.3). Ez az idővonal jól mutatja, hogy a fejlesztők gyorsan reagáltak a robusztusságra és a funkcionalitás bővítésére vonatkozó igényekre. A prototípus tesztelési fázisai a rendszer három kritikus területének validálására összpontosítottak: a kommunikáció megbízhatósága, a távoli firmware frissítés üzembiztonsága, valamint a Field-alapú tartalomkezelés funkcionalitása.

7.2 Tesztelési megközelítés és eredmények:

A tesztelés elsődleges célja volt annak biztosítása, hogy a vezérlő megbízhatóan képes fogadni és végrehajtani a parancsokat, függetlenül a kommunikációs csatorna minőségétől. A zajos környezet (RS485) és a csomagvesztéses kapcsolatok (GPRS) szimulációja során ellenőrizték a csomagkeret integritását, különös tekintettel a CSSZ1 és CSSZ2 kettős ellenőrző összeg pontosságára. A tesztek igazolták, hogy ez a redundáns ellenőrzési mechanizmus hatékonyan kiszűri a sérült csomagokat, minimalizálva az adathibák kockázatát. Emellett a F0h/F1h (Ping) parancs tesztelésével igazolták a kapcsolat élő voltát nagy adatforgalom mellett is. Ezen felül a Field beállítások tömeges küldését végző 64h parancs gyors végrehajtási képességét is validálták.

A firmware frissítés robusztussági tesztelése során szimulálták a tápfeszültség szüneteket és a kommunikációs megszakításokat a frissítési ciklus különböző pontjain (különösen a 0Ch parancs végrehajtása közben, amikor nincs válasz). A rendszer elvárása az volt, hogy minden esetben képes legyen futtatható programot indítani. A teszteredmények megerősítették, hogy a beépített integritás-ellenőrzési mechanizmusnak köszönhetően a vezérlő sikeresen helyreállt a megszakításokból, újraindítva a régi, működőképessé firmware-t vagy sikeresen befejezve az új program írását. A folyamat sikerességét a 16h paranccsal lekérdezhető szoftver verzió számának (Projekt verzió) megfigyelésével, valamint a 10h válaszban lévő Reset számláló ellenőrzésével validálták. Ez a fail-safe mechanizmus a távoli karbantartás egyik legfontosabb pillére.

A kijelzőfunkciók tesztelése során validálták a Field-alapú megjelenítést és az átlapolódás kezelését. Tesztelték, hogy a 80 Field dinamikus definíciója (60h), valamint a Field set begin (72h) és Field set end (74h) parancsokkal történő tömeges frissítés gyorsan és hiba nélkül megtörténik-e. A tesztek igazolták a Field aktiválásának prioritási logikáját: az utoljára aktivált Field kerül felülre, ami egyszerű, de hatékony renderelési szabályt biztosít.



3. Ábra: RGB kijelző teszt közben

A riasztási funkciók validálása a küszöbszintek (pl. belső hőmérséklet: 70 C, 4. paraméter) beállításával és a fizikai behatolás szimulálásával történt. A tesztek megerősítették, hogy az Alert események (pl. Dobozbontás) bekövetkeztekor az eszköz azonnal, de a paraméterlistában beállított türelmi időt figyelembe véve (22. SMS riasztás közötti idő), küld SMS-t az Admin telefonszámra. A proaktív státuszküldési funkciót az E0h és E2h parancsok használatának engedélyezésével validálták, bizonyítva, hogy a vezérlő képes autonóm módon kommunikálni az állapotváltozásokat.

8 Konklúzió

A UniLED Display kijelző vezérlő prototípus fejlesztési projektje sikeresen elérte céljait, létrehozva egy robusztus, hibrid kommunikációs képességekkel rendelkező beágyazott rendszert. A prototípus meghaladja a hagyományos kijelzővezérlők képességeit azzal, hogy integrált távfelügyeleti végpontként működik, teljesítve az Ipar 4.0 környezet elvárásait.

Főbb technológiai eredmények:

- **Univerzális Kommunikáció:** A Master-Slave protokoll egységesen alkalmazható RS485, LAN, és GSM/GPRS csatornákon, növelve a telepítési rugalmasságot.
- **Adatbiztonság:** A kommunikációs csomagok kettős ellenőrző összeggel (CSSZ1 és CSSZ2) való védelme kritikus megbízhatóságot garantál.
- **Üzembiztonság:** A fail-safe firmware frissítési mechanizmus biztosítja, hogy az eszköz bármely kommunikációs megszakítás vagy áramszünet esetén is képes legyen futtatható szoftverrel újraindulni.

- **Dinamikus Field-vezérlés:** A kijelző 80 független Field-re osztható, támogatva a valós idejű adatok (óra, hőmérséklet) és a hosszú, akár 1000 karakteres gördülő stringek megjelenítését.
- **Proaktív Diagnosztika:** Az integrált szenzoros ökoszisztéma (dobozbontás, vízbetörés, túlmelegedés, feszültségmérés) aszinkron riasztási mechanizmusokkal (SMS, E2h) kombinálva teszi lehetővé a proaktív felügyeletet.

Javaslatok a jövőbeli fejlesztésekre:

Bár a prototípus magas érettségi szintet ért el, a hardver és a protokoll már felkészült két olyan jelentős funkció jövőbeli aktiválására, amely növelheti az eszköz piaci értékét:

- **Hanglejátszás funkció bővítése:** A hardveren található microSD foglalat és az AUDIO port jelzi a hanglejátszó modul meglétét vagy beépítésének lehetőségét (3. Opcionális hanglejátszó modul megléte paraméter). A protokoll már tartalmazza az 52h (fájl(részlet) lementése), az 58h (hangfájl lejátszása) és az 5Ah (lejátszás leállítása) parancsokat. A jövőbeli szoftverfejlesztés fókuszálhat a beágyazott MP3 dekódolás és a hangfájlok SD kártyán történő menedzselésének szoftveres aktiválására.
- **HUB08 Támogatás Integrálása:** A nyomtatott áramkörü lap (NYÁK) tartalmaz egy HUB08 típusú csatlakozót (Monokróm kijelzők számára), de a jelenlegi firmware (CS143.8) még nem támogatja ezt az interfészt.¹ A kijelző modul típusának (6. paraméter) bővítése és a hozzá tartozó SW illesztőprogram kidolgozása a jövőben lehetővé tenné a termék szélesebb körű kompatibilitását.

UniLED vezérlő piaci felhasználása projektekben:

A prototípus alapján készült vezérlőket már több projektben felhasználtuk, bizonyítva a koncepció és a prototípus fejlesztés életképességét és működő képességét valós környezetben.

2021 és 2023 évek között Miskolc városában az autóbusz állomás kijelzőit (22 db) szállította cégünk UniLED vezérlővel, valamint mintegy 140 közterületi kijelzőt újítottunk meg és cseréltük a meghajtást UniLED vezérlőre.