SZAKDOLGOZAT

Bognár Botond
Mechatronikai mérnök szak

Gödöllő
2025

MAGYAR AGRÁR- ÉS ÉLETTUDOMÁNYI EGYETEM

Szent István Campus, Gödöllő
Cím: 2100 Gödöllő, Páter Károly utca 1.
Tel.: +36-28/522-000
Honlap: https://godollo.uni-mate.hu

Magyar Agrár- és Élettudományi Egyetem
Szent István Campus
Mechatronikai Mérnök Alapképzési Szak

Célgép PLC programja

Belső konzulens: Tóth János
Egyetemi adjunktus
Külső konzulens: Kulcsár Dávid
Villamosmérnök
Készítette: **Bognár Botond**
YWIKY1
Nappali tagozat
Intézet/Tanszék: **Mechatronika**

Gödöllő
2025

Kitöltött feladatlap

MAGYAR AGRÁR- ÉS ÉLETTUDOMÁNYI EGYETEM

Szent István Campus, Gödöllő
Cím: 2100 Gödöllő, Páter Károly utca 1.
Tel.: +36-28/522-000
Honlap: https://godollo.uni-mate.hu

MŰSZAKI INTÉZET
MECHATRONIKA ALAPSZAK
Gépgyártó specializáció

SZAKDOLGOZAT

feladatlap

Bognár Botond (YWIKY1)

részére

A szakdolgozat címe:

Célgép PLC programja

Feladatkiírás:

Bevezetés, szakirodalom feldogozás, probléma bemutatása, célgép PLC programozása, gazdasági számítás, összefoglalás

Közreműködő tanszék: Mechatronika

Külső konzulens: *Kulcsár Dávid, Ügyvezető,* ElectroZen Kft.

Belső konzulens: *Tóth János, egyetemi adjunktus*, Magyar Agrár és Élettudományi Egyetem, Műszaki Intézet

Beadási határidő: 2025. november 04

Gödöllő, 2025. szeptember 08

Jóváhagyom **Átvettem**

(tanszékvezető) (szakfelelős) (hallgató)

A dolgozat készítőjének külső konzulense nyilatkozom arról, hogy a hallgató az előre egyeztetett konzultációkon megjelent.

Gödöllő, 2025.

(külső konzulens)

Tartalom

Szent István Campus, Gödöllő
Cím: 2100 Gödöllő, Páter Károly utca 1.
Tel.: +36-28/522-000
Honlap: https://godollo.uni-mate.hu

1. Bevezetés

Szakdolgozatom témája a Programmable Logical Controller, PLC, azaz programozható logikai egység programozása. A dolgozatban részletesen kifejtem, hogyan működnek, miért használja az ipar, hogyan lehet más elektromos berendezésekkel használni. A bemutatott program és összeépített gép a valóságban is elkészült és a megrendelő rendszeresen használja.

1.1. Téma jelentősége

A programot és a gépet megrendelésre készítettem. Ez a célgép egy műanyagszárító berendezés vezérlését hajtja végre néhany funkcióval kiegészítve. A cég ahol dolgozom többek közt egyedi gépgyártással foglalkozik, így ezt a gépet is tervezni, kivitelezni, programozni és megépíteni is nekem kellett.

Mivel a relék erre a feladatra nem megfelelőek, hiszen számos relé, hosszas, logikát megvalósító elektromos tervezés és rengeteg kábelezés lenne szükséges ahhoz, hogy hasonló szintű relés vezérlést lehessen megvalósítani. Emellett a relés vezérlés lényegesen lassabban is reagál így a programozható logikai egységekre esett a választásom. A PLC-knek számos előnye van. Konfigurálható a be és kimenetek száma, feladatorientáltan programozható a saját környezetében. Ahhoz, hogy az operátor ki tudja választani a megfelelő programot a tizenkettő közül, szükség volt egy human-machine interface-re is (továbbiakban HMI). Magát a HMI-n futó programot is meg kell tervezni, írni és összekapcsolni a PLC-n futó programmal. A géppel szemben támasztott követelmények adottak voltak, az elkészített berendezésnek meg kellett felelnie ezen követelményeknek, melyeket a 3.1 fejezetben részletezek.

1.2. Célkitűzés

A berendezés elsődleges funkciója, hogy tizenkét fajta műanyagot tudjon szárítani, ezek mind különböző száradási idővel rendelkeznek. Mivel a szárítóberendezésben magas a hőmérséklet, a program indulásával egy mágneses ajtózár zárja az ajtót és látja el a biztonsági funkciókat. Az időzítő lejártával pedig megszólal egy jelző, hogy a folyamatnak vége. A cél egy egyszerű, praktikus, felhasználóbarát felület létrehozása, mely eleget tesz a megrendelő elvárásainak.

2. Szakirodalom feldolgozása

Ebben a fejezetben részletesen tárgyalom a gépben felhasznált elemek fejlődését, működését, illetve a PLC-k és HMI-k működését, fajtáit és programozását.

2.1. PLC története, működése és fajtái

Ebben a fejezetben részletesen tárgyalom a PLC-k fejlődését, működését. Bemutatom milyen fajta PLC-k kaphatóak, milyen kommunikációs módok vannak, majd összehasonlítom a relés vezérléssel.

2.1.1. PLC története

A programozható logikai egységeket először az 1960-as évek vége, 1970-es évek elején kezdték el használni ipari környezetben. [1] Az újítás számos előnnyel (kevesebb vezetékezés, gyorsabb reakcióidő, egyszerűbben módosítható) járt az eddig használt relés megoldással szemben. Ahogy a vezérlési feladat egyre komplexebbek lettek, nehezebb volt a sok relét bekötni, huzalozni, így kellett egy olyan megoldás, ami könnyebben alakítható módosítható utólagosan.

A PLC-k jelenleg már számos feladatot el tudnak látni egy rendszer vezérlésén kívül, többek közt: [2]

- Jelek feldolgozása
- Érzékelők, végrehajtó szervek kezelése
- Kommunikáció más eszközökkel
- HMI
- Dokumentálás, tesztelés
- Tápellátás

Már megjelentek olyan mikroszámítógépek is, melyeken grafikus operációs rendszer (Windows 7/10) fut [3]. Ennek akkor van haszna, amikor a vezérlés mellett meg is kell jeleníteni adatokat, például egy folyamat eredményét, hibákat, elkészült darabszámot vagy egy folyamatjelzőt, esetleg olyan programot kell futtatni, a PLC program mellett, ami csak Windows operációs rendszeren érhető el. Ilyen feladatokra használhatóak a Beckhoff PC-k.

2.1.2. PLC működése és felépítése

Egy programozható logikai egység több részből épül fel. A vezérlési feladatot a központi feldolgozóegység végzi (Central Processing Unit, CPU). Többféle memóriaterület is található az egységben, program és adatmemória. A programmemória nem törlődik egészen addig, amíg új programot nem töltünk a vezérlőegységre, vagy nem formázzuk, viszont az adatmemóriának csak bizonyos részei tartják meg az értéküket áramkimaradás esetén, erre érdemes figyelni a változók deklarálásakor. Ezen kívül van a vezérlőegységnek bemeneti (input) és kimeneti (output) csatlakozófelülete, melyekre fizikailag lehet társítani a programban meghatározott változókat. [2]

A PLC-k programozása több nyelven is lehetséges, ez az IEC 61131-3 szabványba van foglalva, miszerint: vannak szöveges és grafikus rendszerű nyelvek. Grafikus rendszerű nyelveknél a legismertebb a létradiagram (LAD) és a sorrendi folyamatábra (SFC). Szöveges nyelvek közül a legtöbb PLC strukturált programnyelven (ST) írható, emellett elterjedt még az utasítássorozat (IL). [4] Ez alól a Beckhoff PLC-k kivételt képeznek, ott a programozási nyelv sokkal inkább hasonlít egy általános programozási nyelvre, mint például a Python. Szintaktikában nem egyezik meg teljesen a két programozási nyelv, de például be lehet ágyazni Python programokat Twincat rendszerbe. [5]

A programozható logikai egységekben az előre megírt és rátöltött kód fut. Ciklusidőnek azt nevezzük, amikor a kód az elejétől a végéig lefut. Egy ciklus alatt a PLC olvassa a bemeneteket, lefuttatja a programot, vezérli a kimeneteket és kommunikál beállítás szerint más eszközökkel. Ez a folyamat látható az 1. ábrán. [6]

1.ábra: PLC ciklusa

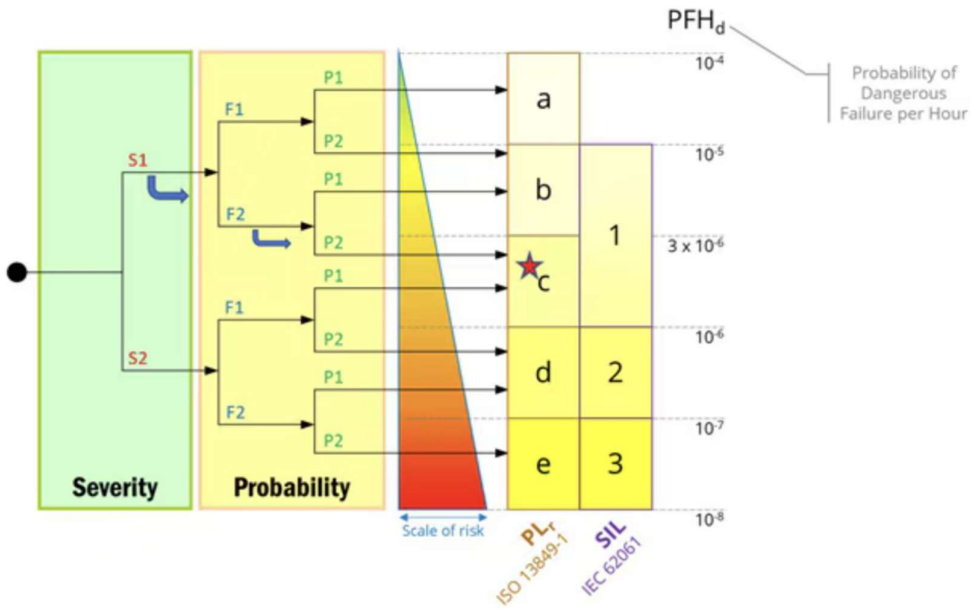
2.1.3. PLC-k típusai

Programozható logikai egységek számos formája fellelhető és alkalmazható ma az iparban. Általam legtöbbe látott és használt PLC az általános PLC, azonban, ezen kívül vannak speciális egységek is, ilyen a biztonsági (safety) és a Beckhoff PLC.

Biztonsági programozható logikai egységek

A biztonsági reléket, mint ahogyan az általános reléket is, leváltották a safety PLC-k. [7] Ez számos előnnyel járt, míg a reléknél a logikát huzalozással kellett megoldani, és az elvégezhető műveletek is korlátozottak voltak veszély esetén. Komplexebb rendszereknél, ami számos relét igényel, akár drágábbra is jöhet ki a sok biztonsági relé, mint egy biztonsági PLC. A biztonsági programozható logikai egységekkel ezek a problémák mind megoldódtak, sőt vezérlési kategóriában is lényeges előre lepések történtek. [8] Elérhetővé vált a PLe, vagyis az ISO 13849-1 szabványban foglalt legmagasabb elérhető vezérlési kategória. A vezérlési kategóriák minimális szintét szabvány írja elő. PL (Performance level) A-tól E-ig terjed PFHd (Probability of Dangerous Failour per Hour) függvényében. Ezeket a kategóriákat mutatja be a 2. ábra. [9]

2.ábra: Performance Level [10]



Szent István Campus, Gödöllő
Cím: 2100 Gödöllő, Páter Károly utca 1.
Tel.: +36-28/522-000
Honlap: https://godollo.uni-mate.hu

2.1.4. Kommunikáció

A modern PLC-k képesek más rendszerekkel kommunikálni, ez ma már több protokoll szerint lehet programozni. Leggyakoribb, hogy az egységek Etherneten keresztül kommunikálnak. Ennek legelterjedtebb formái a: PROFIBUS, CAN, Modbus, CC Link, Ethernet/IP, EtherCAT, TCP/IP. [11] Minden kommunikációs formának megvan az előnye és az alkalmazási területe. Ezen protokollok alapvető összehasonlítása az alábbi táblázatban látható. Néhány gyártó, például a Mitsubishi saját kommunikációs protokollt fejlesztetett ki, mellyel egyszerűbb a gyártó által készített berendezések közti kommunikáció.

1.táblázat: Kommunikációs protokollok

Protokoll	**Típus**	**Valós idejű?**	**Előnyök**	**Hátrányok**
PROFIBUS	Fieldbus	Igen	Elterjedt, szabványos, jó ipari támogatás	Érzékeny kábelezésre, kevésbé rugalmas
CAN	Fieldbus	Igen	Robusztus, egyszerű, olcsó	Alacsony sávszélesség
Modbus	Serial / Ethernet	Nem	Egyszerű, széles körű támogatás	Lassú, nem valós idejű, nem biztonságos
CC-Link	Fieldbus	Igen	Nagy sebesség, Mitsubishi eszközökkel hatékony	Korlátozott kompatibilitás
EtherNet/IP	Ethernet alapú	Részben	Nagy áteresztőképesség, széles ipari támogatás	Hálózati túlterheltség problémás lehet
EtherCAT	Ethernet alapú	Igen	Ultragyors, pontos szinkronizáció	Speciális vezérlők szükségesek

Szent István Campus, Gödöllő
Cím: 2100 Gödöllő, Páter Károly utca 1.
Tel.: +36-28/522-000
Honlap: https://godollo.uni-mate.hu

2.1.5. PLC és relés vezérlés összehasonlítása

A PLC és relés vezérlés összehasonlítása az alábbi táblázatban látható:

2.táblázat: PLC és relés vezérlés

Szempont	Relés vezérlés	PLC-s vezérlés
Felépítés	Elektromechanikus relék, időrelék, kapcsolók	Mikroprocesszor alapú vezérlő, digitális és analóg bemenet/kimenet
Programozhatóság	Nem programozható, huzalozással valósul meg	Könnyen programozható szoftveresen (pl. létra diagram, STL)
Rugalmasság	Nehézkes módosítás – újra kell huzalozni	Nagy rugalmasság – a logika szoftveresen módosítható
Méret és helyigény	Nagyobb helyigény a sok relé miatt	Kompakt, helytakarékos kialakítás
Hibakeresés	Időigényes, manuális vizsgálat	Könnyebb diagnosztika, szoftveres hibakeresés
Bővíthetőség	Korlátozott, új relékkel jár	Modulárisan bővíthető
Karbantartás	Mechanikus alkatrészek kopnak, gyakori karbantartás szükséges	Minimális mechanikai kopás, alacsony karbantartási igény
Sebesség	Lassabb működés (relé behúzási ideje miatt)	Gyorsabb adatfeldolgozás
Költség (kezdéskor)	Alacsonyabb kezdeti költség	Magasabb kezdeti beruházás
Költség (hosszú távon)	Drágább karbantartás és módosítás	Gazdaságosabb hosszú távon a rugalmasabb vezérlés miatt
Alkalmazás	Egyszerű, kis automatizált rendszerek	Bonyolult, összetett vezérlési feladatok

2.2. PLC programozása

A fejezetben tárgyalom, milyen változók, adattípusokkal lehet programozni, mi az a HMI rendszer és hogyan kapcsolódik a PLC-hez.

2.2.1. Adattípusok, változók

Egy PLC kód megírását megelőzően meg kell tenni néhány lépést. Át kell gondolni, mi a program célja, milyen funkciókat kell ellátnia, milyen környezetben használjuk, szükséges e megjelenítő rendszer, milyen alfunkciók érhetőek el, biztonsági intézkedések és még számos másik kérdésre kell tudni a választ ahhoz, hogy jól működő kód legyen a munkánk eredménye.

A vezérlő programozása legtöbbször a gyártó saját szoftverében történik, például a Mitsubishi PLC-k a Melsoft Navigátorban, Beckhoff PLC-k a TwinCat programban. Ezek minden gyártónál valamilyen szinten eltérnek, de a főbb beállítások, paraméterek ugyan azok.

Első lépés a programozásnál a változók felvétele. A változók olyan névvel ellátott memóriaterület, amely adat tárolására szolgál, értéke a program futása során megváltozhat. A változók lehetnek bemenetek, kimenetek, belső segédváltozók vagy állapotjelzők. Ezek mind típusuk, amit mi határozunk meg a deklarálásukkor. Ez meghatározza, hogy milyen értéket tárolnak. A változók típus szerinti csoportosítása a PLC programozásban a következő fő kategóriákba sorolható: [12]

- BOOL – logikai típus
- BYTE / WORD / DWORD / LWORD – 8/16/32/64 bites adatok
- SINT / INT / DINT / LINT – egész szám típusok
- USINT / UINT / UDINT / ULINT – előjel nélküli egész típusok
- REAL / LREAL – valós szám típusok
- TIME / DATE / TIME_OF_DAY / DATE_AND_TIME – idő és dátum típusok
- STRING / WSTRING – karakterlánc típusok
- ARRAY – tömb típus
- STRUCT / USER-DEFINED TYPES (UDT) – felhasználó által definiált struktúrák
- ENUM – felsorolási típusok
- REFERENCE / POINTER – hivatkozások és mutatók

Szent István Campus, Gödöllő
Cím: 2100 Gödöllő, Páter Károly utca 1.
Tel.: +36-28/522-000
Honlap: https://godollo.uni-mate.hu

A fent említett típusok különböző célra szolgálnak attól függően, hogy milyen típusú adatot kell deklarálni, milyen műveletet kell végrehajtani a változókkal és hogy a gép fizikai folyamataihoz hogyan kapcsolódik.

Az legelemibb változó az a BOOL típus, amit logikai igazságértékekhez, tehát 0 vagy 1, igaz, vagy hamis értéket tud felvenni, ilyen például egy villanykörte, ami vagy világít, vagy nem. Ez a változót például érzékelő állapotai vagy motorvezérlő kimeneti értékek tárolásához használják. Emellett a BYTE, WORD, DWORD és LWORD típusokat bitek csoportos eljárására használják. Ezek a típusok több BOOL változóból tevődnek össze, például egy BYTE az 8 BOOL értéket foglal magába.

A SINT (8 bit), INT (16 bit), DINT (32 bit) és LINT (64 bit) típusoknak előjeles egész számot tartalmaznak, míg a megfelelőik ezeknek előjel nélküliek, azaz USINT, UINT, UDINT, ULINT, akkor érdemes használni, ha csak pozitív értékekre van szükség pl. számlálókra vagy a sorszámokra.

Az REAL, LREAL típust valós számok kezelésére használjuk. Hasonló például az alábbi feladatok kezeléséhez, mikor mondjuk hőmérsékleti, nyomási vagy sebesség értéket kezeltünk. A fenti típusokat nem lehet elhagyni a számítógépes feldolgozásnál.

Az időalapú típusok, mint például a TIME, DATE, TIME_OF_DAY, DATE_AND_TIME lehetőséget nyújtanak az események időzítésére, időbélyegzésére és ciklusidők mérésére. Az időalapú logika célgépek automatizálásában is kiemelt szerepet játszik, legyen szó akár töltési ciklusokról, késleltetésekről vagy ütemezett karbantartásról.

A STRING és a WSTRING típusokat szöveges adatok kezelésére használják. Ezek különösen értékesek akkor, ha a PLC HMI felülethez kapcsolódik, és üzeneteket, hibakódokat vagy paraméterneveket kell megjeleníteni.

Ha több egynél ugyanaz az adattípus adatot kell kezeljük, a tömb típus, azaz ARRAY használata célszerű. Ez egy olyan változó, mely mondjuk nyolc darab WORD típusú változót tárol el egymás utáni címeken. Példának, tíz szenzor jelét szeretnénk észlelni és feldolgozni, tömbbe összeállítva jobb elérése, egyszerűbb kezelése és iterálása lehet ezeknek az értékeknek, mindamellett, hogy a kód lényegesen átláthatóbb.

Szent István Campus, Gödöllő
Cím: 2100 Gödöllő, Páter Károly utca 1.
Tel.: +36-28/522-000
Honlap: https://godollo.uni-mate.hu

A STRUCT típus, másnéven szerkezet megengedi a különböző típusú változók logikai egységbe szervezését. Például egy henger állapotát reprezentáló struktúra már tartalmazhat logikai értéket a végállás érzékeléséről, egy hibakódot egész számként, valamint az utolsó aktiválás idejét idő típushoz. Ez az adatszervezési forma javítja a kód olvashatóságát és utólagos átláthatóságán.

Az ENUM, vagy felsorolás típus lehetőséget nyújt előre definiált, elnevezett állapotokkal való munkára. Ez inkább állapotgépek esetében működik, ahol például a gép állapotát tárolhatjuk el a változóban és ezt jelezhetjük ki egyszerűbben, például: áll, fut, auto, manual, hiba, vészleállás.

A referencia típusú változók, mint a REFERENCE vagy a POINTER, vezető szintű programozáskor használjuk. Rájuk hivatkozva dinamikus hivatkozást tehetünk más változókra vagy adatszerkezetekre, ami pl. paraméterezhető funkcióblokkok kódolását vagy memóriahasználatot optimalizáló képességgel ruházza fel a kódot. [13]

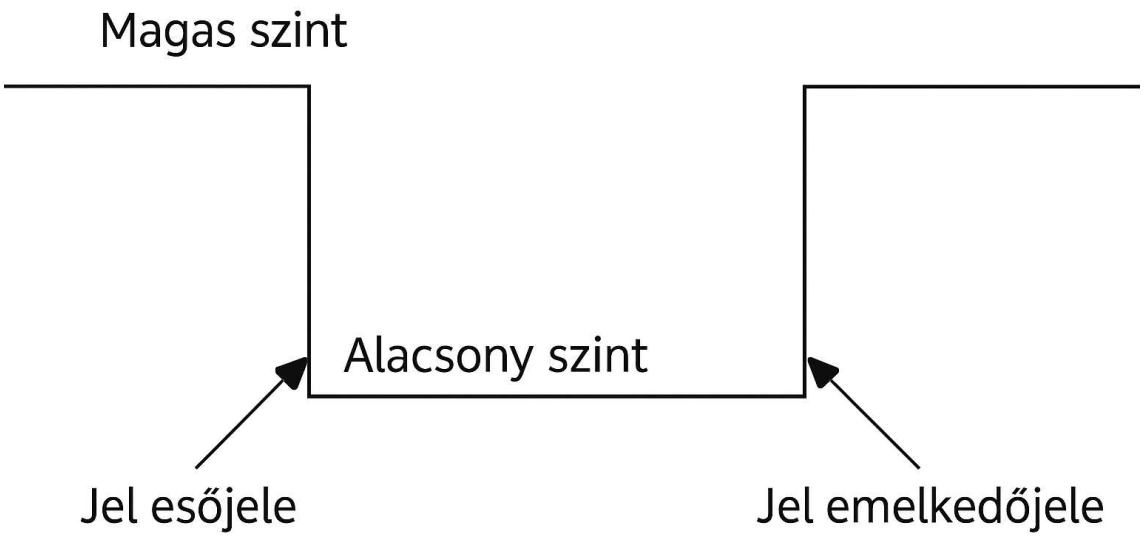
A változók típusa megfelelő kiválasztása elengedhetetlen a program teljesítményének, karbantarthatóságának és hibamentes futásának szempontjából. A célgépek programozásánál kifejezetten körülményes, hogy a fizikai jelekhez, folyamatparaméterekhez és vezérlési logikához illő típusokat használjunk, és gondosan strukturáljuk a programkódot.

Szent István Campus, Gödöllő
Cím: 2100 Gödöllő, Páter Károly utca 1.
Tel.: +36-28/522-000
Honlap: https://godollo.uni-mate.hu

2.2.2. Élvezérlés

A hagyományos PLC-programok ciklikus működése során a vezérlő minden egyes ciklusban lefuttatja a rátöltött kódot, függetlenül attól, hogy a bemenetek állapota változott-e vagy sem. Ez az eljárás egyszerű és időfüggetlen, de sokszor nem elég, ha csak a pillanatnyi jel-állapotra támaszkodunk. Néhány alkalmazásnál, például egy nyomógomb megnyomásakor pont az számít, hogy egy jel mikor vált állapotot, nem pedig az, hogy milyen értéken áll épp. Ilyenkor jön képbe az élérzékeny logika. [14]

3.ábra: Élvezérlés



Digitális bemenetek esetén a jel alacsony (0) és magas (1) szintje közti átmeneteket jelleg szerint kétféleképpen nevezzük, ahogyan a 3.ábrán is látható:

- felfutó él: amikor a jel 0-ról 1-re vált
- lefutó él: amikor a jel 1-ről 0-ra vált.

Az élváltozásra történő vezérlés célja, hogy nem a jel szintjére, hanem az élen bekövetkező váltására reagáljon. Így kiküszöbölhetők az ismétlődő vagy fölösleges programlépés, különösen akkor, ha egy bemenet hosszabb ideig aktív marad.

Erre egy példa a berendezéseken gyakran használt indító gomb. Amint lenyomjuk a gombot, a bemeneti jel magasra ugrik, és csak a felengedéskor tér vissza alacsonyra. Ha a vezérlő minden ciklusban detektálná a magas szintet, ugyanaz a művelet többször is elindulna, amíg a gomb nyomva van tartva. Élérzékeny logika esetén viszont csak a felfutó élre lépünk tovább: a hozzá tartozó utasítás egyszeri kiváltása a gomb első megnyomásakor történik.

A legismertebb IEC 61131-3 [13] szerinti nyelvek mind támogatják az élérzékeny működést, de eltérő módon:

Például ST-ben gyakoriak az R_TRIG (felfutó él) és F_TRIG (lefutó él) blokkok. Ha ilyen nincs előre definiálva, a programozó minden ciklusban elmenti a jel előző értékét, és a mostani állapottal való összehasonlítással állapítja meg az élváltást. [15]

Az élérzékeny logika előnyös, ha a folyamat egy sor lépésre van bontva, események által indított állapotgépként működik. Például:

1. A kezelő behelyezi az alkatrészt és megnyomja az indítógombot.
2. A következő lépés csak akkor indul, ha az érzékelő igazolja a helyes pozíciót.
3. Ezután a mozgatómechanika végálláshoz ér, ami ismét élváltást generál.

Minden lépésváltás egy él megváltozása indít, ezért az élérzékeny vezérlés megbízható. A korszerű vezérlőrendszerek gyakran alapból módon támogatják az élérzékeny funkciókat. A fejlettebb PLC-k előre elkészített élvezérlésre alkalmas blokkokat kínálnak, így a programozás gyorsabb, átláthatóbb és kevesebb hibalehetőséget rejt.

Az élérzékeny logika ma már nem csak kiegészítő elem, hanem fontos része a célgépek vezérlésének. Lehetővé teszi, hogy a rendszer kizárólag a valódi eseményekre reagáljon, minimálisra csökkentve a fölösleges ciklusokat, növelve a működés pontosságát és megbízhatóságát. [16]

2.2.3. HMI rendszerek

A HMI, azaz a Human–Machine Interface rendszerek fontos szerepet játszanak az új ipari automatizálásban. [17] A céljuk, hogy átmentet képezzenek az emberi kezelő és a gépi folyamat között, megnyitva a rendszer állapotának valós idejű megjelenítését, a paraméterek módosítását, a vezérlési parancsok kiadását és a hibaüzenetek kezelését. A HMI nem csak egy

egyszerű kijelző vagy kezelőpanel, hanem egy komplex kommunikációs és felügyeleti rendszer, amelynek segítségével a kezelőbiztonság, a folyamatellenőrzés és a hatékonyság jelentősen javítható. [18]

A HMI-k fejlődése a korábbi, egyszerű gombnyomógépektől a jelen grafikus, érintős, hálózatba kötött rendszerekig ível. Régebben a kezelő [19] egy LED-es és egy kapcsoló segítségével jelezte a gép állapotát; ma pedig bonyolult kezelőfelületen élőben követünk folyamatokat, grafikonon kijelölt adatot, paramétereket. Hibakezdetkor rendszeresített hibabejelentést vagy diagnosztikát visszajelzésként is készíthetünk A HMI nem csak egy folyamat irányításában segít, hanem karbantartáskor is szükséges: ha egy túlmelegedés vagy nyomás-ingadozás során lép föl például a kezelő egyből látja az adott pillanat értékét, és ha van rá szükség, beavatkozhat.

Főbb jellemzője a HMI-rendszereknek a könnyű kezelhetőség, ahol az ember-gép kapcsolat gördülékenyen, hiba nélkül működik. Mindemellett egy HMI teljes mértékben testre szabható, hogy a felhasználó a lehető legkönnyebben használhassa a számára készített rendszert. Fontos azt is megeml0teni, hogy a HMI szoftver és hardvernek meg kell felelnie néhány szabványnak, hogy megfeleljenek a biztonsági és funkcionalitási elő0rásoknak: többek közt: IEC 61131-3 [13], IEC 61508 [20], ISO 13849 [21]. A HMI fő funkciói közé tartozik:

- **Folyamatvizualizáció**
  A kezelő számára valós idejű visszajelzést nyújt a folyamat aktuális állapotáról. Ez lehet színes grafikus ábra, gépstátusz, paraméterek, üzemmód kijelzése, jeltartomány (pl. hőmérséklet) trendje
- **Paraméterezés és vezérlés**
  A paraméterek beállítása, módosítása, mérési tartományok vagy határértékek beállítása érintőképernyőn vagy billentyűzetről, továbbá kezelői utasítások küldése a vezérlőrendszer felé.
- **Riasztás és hibakezelés**
  Ha egy folyamatüzem hibát, túllépést, üzemzavarokat észlel (például túl magas nyomás), a HMI azonnal riasztást ad, megjeleníti a hibaüzenetet, időbélyeggel látja el, és segíti a hiba okának felderítését

- **Adatgyűjtés és naplózás**
  Ciklikusan vagy eseményvezérelten rögzíti a mért jellemzőket, üzemmódokat, státuszokat, amelyek később riportokban, visszakeresésekben, teljesítményösszehasonlításokban használhatók
- **Karbantartás támogatása**
  Hibák diagnosztizálása, alkatrészcserék dokumentálása, szervizsémák és eljárások egyszerű elérése, esetenként távoli támogatás – beleértve akár távoli hozzáférési módokat (VPN, webes felület)

A fenti funkciók összessége alkotja az HMI rendszer átfogó képességeit. Minden elem, legyen az grafikus megjelenítés, riasztáskezelés vagy adatgyűjtés, az együttműködés alapjára épül, melynek az eszközök közötti megbízható kommunikáció az alapja. Ez számos ipari kommunikációs protokoll támogatását igényli, mint például az OPC UA, Modbus, ProfiNet, EtherNet/IP, CANopen, BACnet vagy MQTT – különösen a IoT (Internet of Things) és az ipari digitalizáció korában. [22]

Az emberi interfész mellett nem elhanyagolható a kezelői élmény. Egy jól megtervezett HMI felület:

1. provokálja a kezelő figyelmét, ha baj van;
2. egyszerűvé teszi az interakciót – akár érintéssel, akár billentyűzettel;
3. segíti a kezelőt a folyamat során hibamentesen végig menni az eljáráson;
4. támogatja az oktatást és betanítást, hiszen a jól megtervezett kezelőfelület önmagában oktató jellegű is lehet.

Ha ugyanez a rendszer nem intuitív vagy összetett, az emberi hiba lehetősége megnő. Ez különösen érzékeny, minőségi vagy biztonságkritikus környezetekben.

Grafikus megjelenítés terén elterjedtek a színes, animációs elemek, trendgrafikonok, interaktív diagramok, állapotjelző fények, LED-sorok és státuszsávok.
Ezek lehetővé teszik a kezelő számára, hogy gyorsan átlássa a rendszer állapotát, és gyorsan reagáljon szükség esetén. Például egy animációs motor képes megmutatni a mechanikai mozgás irányát, ha pedig túlterhelés lép fel, piros színnel jelzi a veszélyt. Ezzel a látványos eszközzel az információ gyorsan és közérthetően jut el a kezelőhöz, ezzel csökkentve a felesleges időt.

A HMI hardverének típusai a következők lehetnek: [23]

- **Panel PC-k**, amelyek nagyméretű érintőképernyős eszközök, általában x86-alapú ipari számítógépek. Ezek alkalmasak bonyolult grafikai megjelenítésre, riportkészítésre, és gyakran fut rajtuk teljes operációs rendszer (Windows Embedded, Linux).
- **IP67-es kezelőpanelek**, amelyek por- és vízálló kivitelűek, strapabíróak, ipari környezethez szabottak, és érintésérzékenyek.
- **Kompakt CPU+HMI eszközök**, ahol PLC és HMI egy dobozban van – költséghatékony és egyszerű telepítést kínálnak.
- **Webes HMI-k**, ahol a felület böngésző alapú, és akár mobilon vagy táblagépen is elérhető – különösen hasznos a távoli monitoringhoz és mobil irányításhoz.

A HMI-szoftverek számtalan céleszközre épülnek. Számos ilyen platform található, mely közül néhány:

- Siemens WinCC;
- Rockwell FactoryTalk View;
- Schneider Vijeo Designer, EcoStruxure;
- Beckhoff TwinCAT HMI;
- Indusoft Web Studio;

Ezek között vannak szigorúan zárt platformú, multinacionális gyártói megoldások, de elérhetők nyíltabb, skálázható rendszerek is.

A HMI-k fejlesztése során kiemelt figyelmet kell fordítani a biztonságra. A kritikus rendszerek bizalmas folyamatai veszélybe kerülhetnek egy esetleges behatolás miatt. Ezért a HMI-szoftverek:

- hitelesítési és autorizációs mechanizmusokat kell, hogy tartalmazzanak (jelszavas belépés, jogosultsági szintek);
- kommunikációjuk titkosított (TLS/SSL, VPN) legyen;
- auditnaplókat vezessenek az eseményekről (ki mit és mikor módosított);
- rendszeresen teszteljenek sebezhetőségeket

A HMI környezete sem működik önmagában: háttérrendszere, adatbázisok (Structured query language, SQL) integráltan kapcsolódnak hozzá. Ilyen esetekben a HMI nem csak megjelenít, hanem adatokat szolgáltat, vezérlést indít, riportokat készít, és visszacsatolásokat tesz lehetővé. Elengedhetetlen, hogy mindkét irányban biztosított legyen az adatkommunikáció.

Az emberi–gépi felület jövője az IoT és az ipar 4.0 irányába mutat. A felhasználók mobil eszközökkel is hozzáférhetnek a HMI-hez, akár felhőszolgáltatáson keresztül. A mesterséges intelligencia által generált javaslatok, kognitív figyelmeztetések és automatizált vizualizációk új kapukat nyitnak meg. A gépi tanulás és az előre jelző karbantartás integrációja révén a HMI nemcsak passzív megjelenítő, hanem aktív döntéstámogató eszközzé válik.

2.3. Kismegszakítók története és működése

Az első kismegszakító 1923-ban jelent meg, a STOTZ cég fejlesztette ki. [24] A fejlesztés elsődleges célja az olvadóbiztosítékok lecserélése és a hálózatok biztonságosabbá tétele volt.

A kismegszakítók nevükből adódóan a kisebb feszültségek megszakítására alkalmasak, túlfeszültség vagy zárlat esetén kapcsol és megszakítja a hálózatot. Felhasználásuk a háztartásoktól elkezdve az ipari létesítményekig mindenhol elterjedt. A kismegszakítók elterjedt alkalmazása miatt, ma már tömeggyártásban készültnek, ezért anyaghasználatát és szerkezetét is ennek megfelelően kellett kialakítani, melyet a MSZ EN 60947/2 [25] számú szabvány rögzít.

A kisfeszültségű megszakító működését egy bimetál szalag biztosítja, mely túláram esetén melegszik és ha átlépi a gyártástechnológiával beállított értéket, akkor egy mechanikus kapcsoló lép működésbe és szakítja meg az áram folyását. [26] Ahhoz, hogy az áram újra tudjon folyni a kisfeszültségű megszakítón keresztül, manuálisan vissza kell állítani a kismegszakítón lévő kapcsolót a megfelelő pozícióban. [27]

Működésük alapján számos kisfeszültségű megszakítót különböztethetünk meg. Termikus, amely a túláram által kialakult hőmérsékletkülönbségre kapcsol, mágneses, amely a vezetékek körül kialakult megnövekedett mágneses tér hatására szakítja meg az áramkört. Elterjedt még a hibrid, mely az előbb említett kétfajtának az együttesen beépített alkalmazása. Illetve, néhány alkalmazási terület pontosabb kapcsolást igényel, ilyen alkalmazható az elektromos relé, ami a pontosság mellett még gyorsabb is mint az eddig említett típusok. A differenciális áram megszakítók pedig a föld és rövidzárlatok ellen okoz kiváló védelmet. A kismegszakítókat

védelem szerint is lehet kategorizálni, itt A-tól K-ig, ahol az A típusú olyan alkalmazása javasolt, ahol ismert a maximum feszültség és az nagyjából állandó, a K típusú pedig olyan területre javasolt, ahol gyakran történik zárlat vagy túláram. [26]

Szerkezeti felépítését tekintve a kisfeszültségű megszakító két fő csoportra oszthatóak: [24]

- Kapcsolószerkezet: áramvezető részek, érintkezők, csatlakozókapcsok
- Kioldó: ha az áram meghalad egy adott értéket bizonyos ideig a kioldó kapcsol és megszakítja a hálózatot. A kioldó határozza meg a berendezés jelgörbéjét.

A kismegszakítók legfontosabb jellemzői: [24]

- Pólusszám
- Névleges üzemi feszültség ajánlott értékei (gyártói)
- Névleges áram javasolt értékei
- Névleges frekvencia
- Gyorskioldó kioldó áramtartományai
- Névleges zárlati kapcsolóképesség
- Üzemi zárlat kapcsolóképesség
- Áramkorlátozó képesség
- Mechanikai és villamos élettartam

2.4. DC Tápegységek története és működése

A tápegységek a modern teljesítményelektronika egyik fő elemét képzik. Számos alkalmazási területe van, egészen az asztali számítógépektől az ipari gépekig. Az első DC erőművet Thomas Edison hozta létre 1882-ben New York-ban. Innentől kezdve folyamatos fejlődésnek indultak a tápegységek, míg el nem érték a ma is használt formájukat. Ma már többféle tápegységet különböztetünk meg [28], legelterjedtebb a kapcsolóüzemű tápegység, de gyakran használják az impulzusos tápegységet is.

Szent István Campus, Gödöllő
Cím: 2100 Gödöllő, Páter Károly utca 1.
Tel.: +36-28/522-000
Honlap: https://godollo.uni-mate.hu

2.5. Mágneses biztonsági ajtózárak

Az ipari környezetekben a gyorsan mozgó alkatrészek és a magas nyomás, hőmérséklet okozta kockázatok csak akkor tarthatók teljes mértékben ellenőrzés alatt, ha az elektromos vezérlés mellett fizikai biztonsági elemek is gondoskodnak a dolgozók védelméről. A mágneses ajtózárak ebben játszanak szerepet. Működésük az elektromágnes és egy vaslemez közötti vonzóerőn alapul. Amíg a vezérlőrendszer programozott logikája szerint a berendezés forgó vagy egyéb veszélyes mozgásokat végez, áramot tart a tekercsen és az ajtó zárva marad. Csak a motor leállását és az utómozgások megszűnését követően szűnik meg az áramellátás, és nyitható ki biztonságosan a védőburkolat. [29]

A mágneses zárak tervezése során figyelembe veszik a tartóerőt, ami például a Keyence 500–1000 newton között mozog, így még nagyobb ajtókat is biztonsággal lezár. A visszajelző érzékelők lehetnek reed-relések, induktív adók vagy kódolt jeleket szolgáltató komponensek. Ezek garantálják, hogy a vezérlés pontosan lássa, mikor záródott, illetve mikor oldott a retesz. A poros vagy nedves környezetekben az IP [30] (Ingress protection) védettség kritikus, és számos gyártó kínál különböző fokozatú szigeteléssel rendelkező modelleket. Emellett elengedhetetlen az EN ISO 14119 szabvány szerinti kialakítás, amely meghatározza a mechanikai, elektromos és a manipuláció elleni védelmi követelményeket.

A mágneses ajtózárak előnye, hogy automatikusan oldanak, amint a rendszer feltételei teljesülnek, és nem igényelnek mechanikus karbantartást, így csendesek és hosszú élettartamúak, például a Keyence 1 millió ciklus vagy több mechanikai élettartalmat ígér. [31] Ugyanakkor áramkimaradás esetén oldódnak is, ezért kritikus üzemeknél gyakran alkalmaznak rugós reteszt vagy dupla tápellátást. A leggyakoribb felhasználási területek közé tartoznak a gyártósori védőburkolatok, CNC-ajtók, robotcellák és célgépek kezelőpontjai, ahol a dolgozó közelsége miatt szigorú hozzáférés-szabályozásra van szükség.

Az ajtózárak által nyújtott fizikai védelem azonban csak része a komplex biztonsági struktúrának. gyakran együtt használjuk őket fényfüggönyökkel, vészleállítókkal és kétkezes indítási funkcióval, de néhány berendezésnél, például ennél a műanyag szárítónál is csak önmagában egyedül használom. A jövő felé nézve ezek az eszközök egyre inkább belépnek az ipari IoT-világába: [32] valós idejű állapotfigyelés, prediktív karbantartás és távfelügyelet segítségével még hatékonyabb üzembiztonságot tesznek lehetővé.

Szent István Campus, Gödöllő
Cím: 2100 Gödöllő, Páter Károly utca 1.
Tel.: +36-28/522-000
Honlap: https://godollo.uni-mate.hu

3. Anyag és módszer

Ebben a fejezetben ismertetem a felhasznált szoftvereket és megoldásokat. A projekt jellegéből kifolyólag néhány eszköz, amit használni kellett adott volt. Ezek közé tartozott a PLC, ami Mitsubishi FX3G-24M gyártmányú. Ez meghatározza azt, hogy milyen programot kellett használni a kód írásához, ugyanis Mitsubishi PLC-re csak a saját szoftverével lehet programot írni. A rendszerhez tartozik kijelző is, HMI, ezért több alkalmazás használatára volt szükség.

3.1. PLC technikai adatai

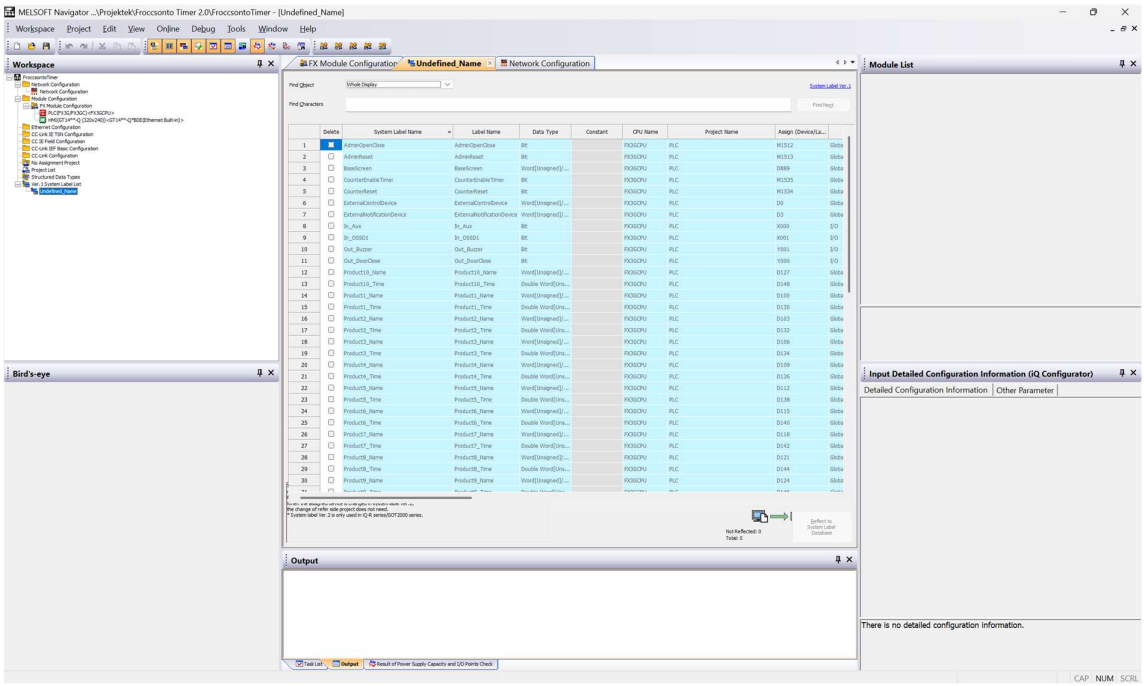
- Pontos típus: Mitsubishi FX3G-24
- Megjelenés éve: 2010
- Áramellátás: 24V DC
- Programmemória: 32.000 lépés
- Bemenetek száma: 14
- Kimenetek száma: 10

4.ábra: Mitshubishi PLC [36]

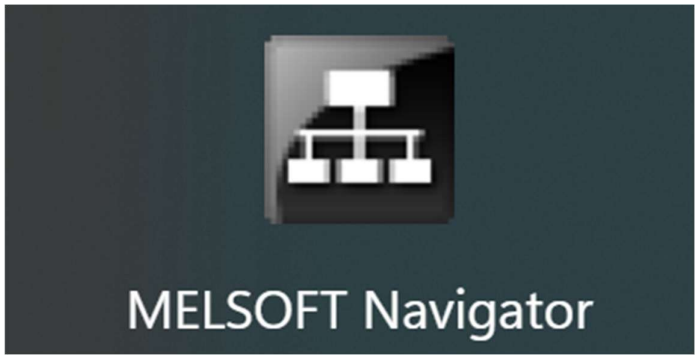
3.2. Felhasznált programok

A MELSOFT Navigator lehetővé teszi, hogy az egész automatizálási rendszer (PLC, HMI, hajtásvezérlő stb.) egy projektben legyen kezelve. Ez átláthatóbbá és egyszerűbbé teszi a vezérlőrendszer felépítését, kapcsolatát és kezelését az egyes eszközök között. Ez az alkalmazás köti össze a PLC-t és a HMI-t. Ennek a segítségével tudjuk a változókat szinkronizálni és a kommunikációt beállítani. Újabb rendszereknél az alkalmazás használata már nem kötelező. Utólagos paramétermódosításnál lehetőség van bármelyik eszköz paraméterét módosítani anélkül, hogy a hozzá tartozó szoftvert meg kelljen nyitni.

5.ábra: Melsoft Navigator program

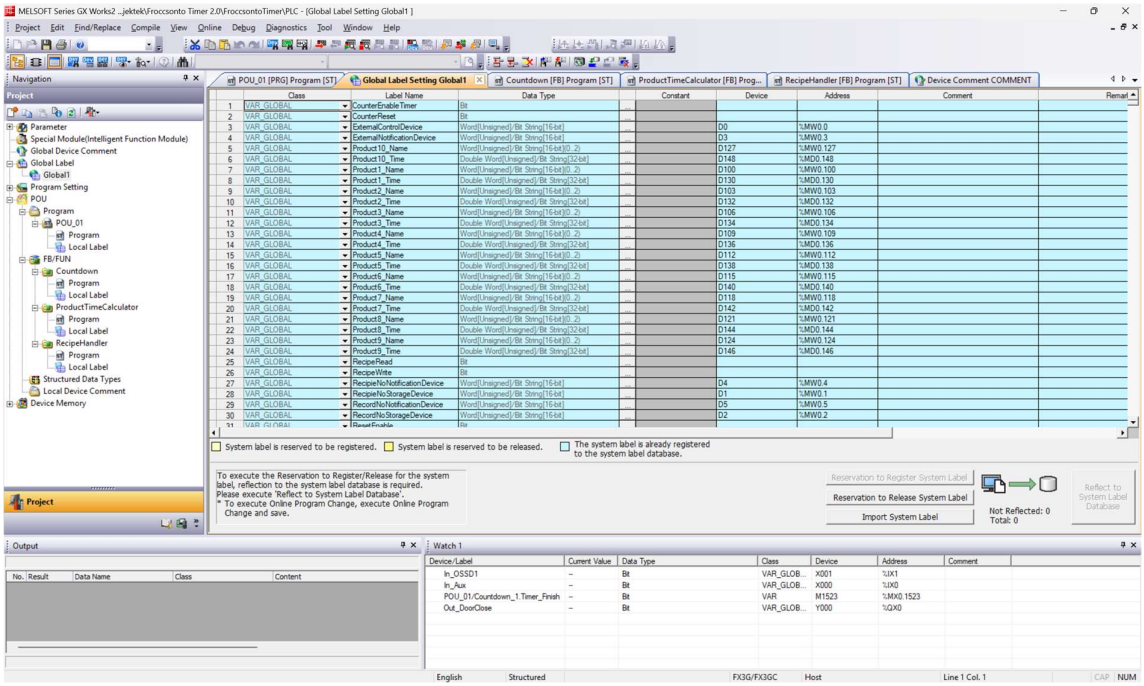


6.ábra: Melsoft Navigator



GX Works 2: Ebben az alkalmazásban lehet a PLC-t programozni. Itt kell felvenni a változókat, funkcióblokkokat létrehozni, és minden egyéb működést leprogramozni, ami nem a grafikus részhez tartozik. A szoftver lehetővé teszi több eszköz és programrész egy projektben való kezelését. Az alkalmazás fő funkciói közé tartozik az online monitor, aminek segítségével a hibakeresés lényegesen egyszerűsödik. Amikor létrejött a PLC-PC kapcsolat és a PLC-n fut a kód, a szoftverben elérhető az online rész, ahol nyomon követhető a változók értéke, aktuális állapot és az esetlegesen fellépő hibák. Ahogyan az ábrán is látható, bal oldalt lehet navigáli a programrészek, funkcióblokkok közt, jobb oldalon pedig megjelenik az éppen aktuális ablak a kóddal vagy éppen a globális változók listájával.

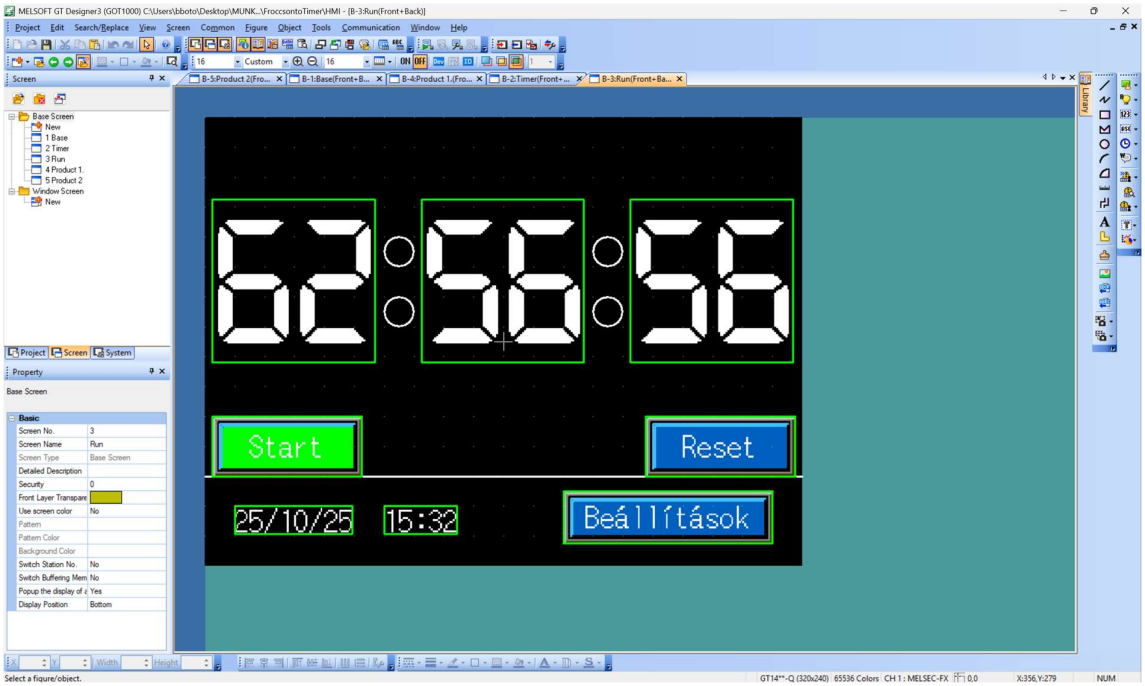
7.ábra: GX Works 2 program



8.ábra: GX Works 2 parancsikon

GT Designer 3: Ez a program tartozik a HMI rendszerhez. A kijelzők és grafikus elemek létrehozásához elengedhetetlen a program használata. A PLC-ben létrehozott változókat itt tudjuk rákötni virtuális gombokra, visszajelző fényekre. Ezen funkciók mellet ebben a szoftverben van lehetőség receptkezelésre, mely tárol bizonyos változókhoz rendelt értékeket. Riasztási és értesítési beállításokat, illetve felhasználókezelést is az alkalmazásban lehetséges beépíteni a rendszerbe. Ahogyan a GX Works 2-ben, itt is lehetőség van tesztelésre, laptopon képes az alkalmazás szimulálni a HMI-t. Mint GX Works 2-ben, és a Navigatorban, itt is bal oldalt található a meüsor és alétrehozott kijelzők. Jobb oldalt itt található egy grafikus eszköztár, innen lehet az elemeket a kijelzőre szerkeszteni.

9.ábra: GT Designer 3 program



10.ábra: GT Designer 3 parancsikon

Szent István Campus, Gödöllő
Cím: 2100 Gödöllő, Páter Károly utca 1.
Tel.: +36-28/522-000
Honlap: https://godollo.uni-mate.hu

4. Eredmények

Ebben a fejezetben bemutatom, hogyan oldottam meg a vezérlési feladatot, milyen HMI-t készítettem és hogyan működik a szoftver. Elvégzem a szükséges számításokat a berendezés összeépítéséhez.

4.1. Működtető szoftver

Első lépéskét elkértem a követelményrendszert, hogy pontosan milyen feladatot fog ellátni a rendszer, milyen fizikai jeleim vannak és milyen funkciók szükségesek. Az alábbi felsorolásban foglalom össze a követelményrendszert, előtte viszont röviden összefoglalom a feladatot:

Műanyag szárítórendszer vezérlése, melybe többfajta műanyag kerülhet és ennek függvényében változik a szárítási idő. A szárítási időt kijelzőről magasabb felhasználóként lehet állítani, illetve a programot folyamat közben is meg lehessen állítani. A rendszerbe tartozik egy mágneses ajtózár is, mely folyamat induláskor bezár, lejártakor kinyit. Magasabb felhasználó ezt is nyithatja-zárhatja folyamattól függetlenül.

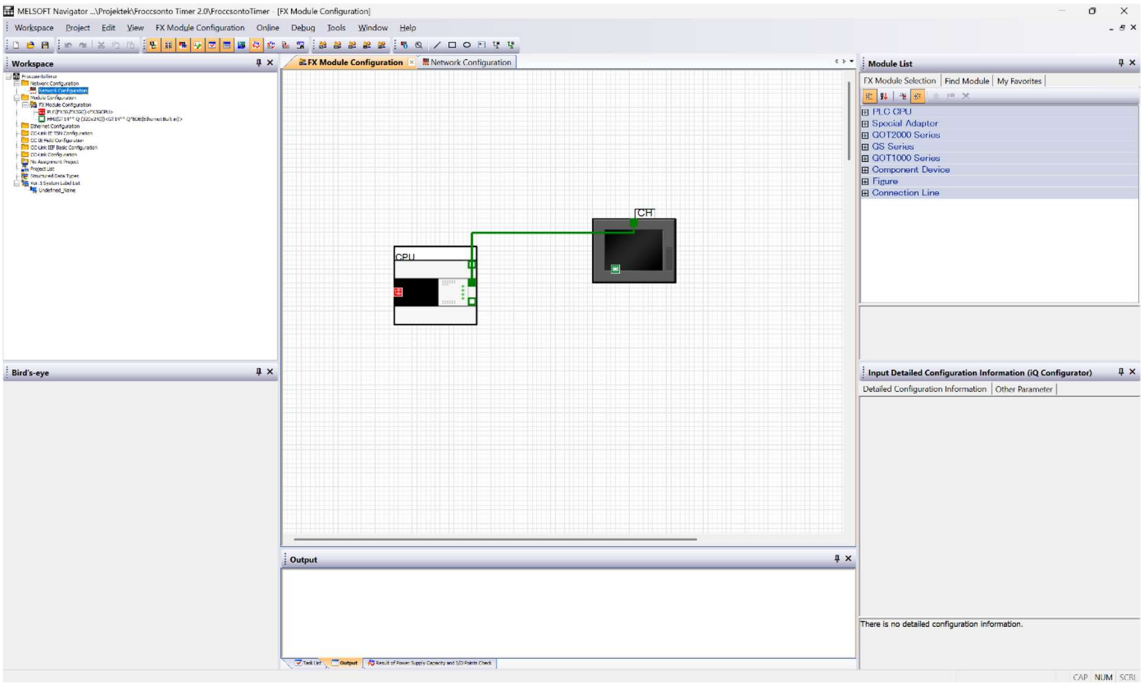
Tehát a követelményrendszer:

- Felhasználókezelés
- Ajtózár működtetése
- Állítható időzítő
- Többféle termék választás
- Felhasználóbarát egyszerű felület
- Hangjelzés a folyamat végéről

A követelmények tisztázása után megnyitottam a Navigator nevű programot és létrehoztam benne a PLC és HMI projektet. Az elnevezések után beállítottam a kommunikáció fajtáját. Ez a kommunikáció soros porton keresztül történik. Modernebb rendszerekben ez már elavultnak számít és Ethernet kábelt használunk. Létrehoztam egy ’FX Module Configuration’-t, majd

ezen belül definiáltam a PLC-t és HMI-t. Itt rendeltem hozzá egyes eszközökre a hozzájuk tartozó programkódot is. Ez látható az alábbi ábrán.

11.ábra: PLC-HMI kapcsolat



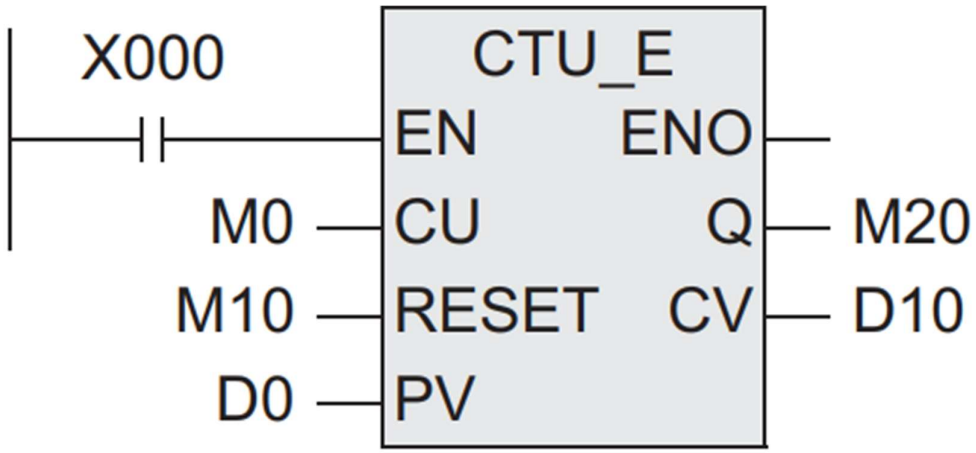
4.1.1. PLC program

A programok hozzárendelése után, megnyitottam a GX Works 2 alkalmazást és létrehoztam a fő programot (Main). Átgondoltam milyen változókra lesz szükségem. Létrehoztam a globális változókat. Ezek azok a változók, amelyeket használni fogok a kijelzőn, illetve azok a fizikai jelek, amelyek beérkeznek a PLC-be, illetve a PLC adja ki őket, ilyen például az ajtózárnak a zárása vagy annak a visszajelzése, hogy zárva van. Ebben az esetben globális változóból 66db van. Leginkább a termékekkel kapcsolatos változók (termék neve, szárítási idő órában és percben), de található több gomb is, például számláló nullázása, ajtó nyitása.

A változók deklarálását követően létrehoztam az első funkcióblokkot, mely magáért az időzítőért volt felelős. Ennek a programrésznek több bemenő feltétele is van. Egyrészt a számláló csak akkor indulhat el, ha az ajtó zárva van és visszaérkezett a bezárva jel, az operátor választott terméket, tehát a termék neve és szárítási ideje betöltődött, és végül pedig rányomott a start gombra, ami elindítja a folyamatot. Ekkor az ajtózár bezár (600-800N kell a

kinyitásához) és elindul a visszaszámolás. A visszaszámlását úgy oldottam meg, hogy a termék ideje, ha mondjuk 2,5 óra, átváltottam másodpercbe, ami ugye 9 000 másodperc, majd egy beépített funkcióblokk és egy olyan belső változó segítségével, ami másodpercenként egy ciklusig magas létrehoztam egy lefelé számláló függvényt. Tehát ha van egy adott idő a függvény minden olyan pillanatban, amikor a belső változó magas levon egyet az értékből és elmenti az egy változóba. A következő ciklusban már ezt a változót tölti be és így folytatja tovább a folyamatot. [33] A változót minden egyes levonásnál felülírja. Ez a beépített funkcióblokk így néz ki:

12.ábra: Számláló blokk [34]



Ahol:

- EN: engedélyező jel a számlálás kezdetéhez
- CU: jel, melyre a számlálás történik
- RESET: számláló újraindítása
- PV: érték, melyet a számlálónak el kell érnie
- ENO: folyamat állapota (aktív, nem aktív)
- Q: magas lesz, ha lejár a számláló
- CV: számláló jelenlegi állapota

Ez a PLC-ben lényegesen egyszerűbb, gyakorlatilag egy sor az egész számláló: `CTU_2(CU:= CounterEnableTimer ,RESET:= Reset ,PV:= Time_Left ,Q:= Timer_Finish ,CV:= Time_Elapsed );`

Szent István Campus, Gödöllő
Cím: 2100 Gödöllő, Páter Károly utca 1.
Tel.: +36-28/522-000
Honlap: https://godollo.uni-mate.hu

Mivel a kijelzőn használok gombokat, azokat felfutó/lefutó élre kell programoznom, ugyanis, ha nem így történne, a program addig indulna el újra és újra, amíg a felhasználó el nem engedi a gombot. Ezekre is található beépített funkcióblokk, felfutó élre az LDF nevű, lefutóra pedig az LDP nevű. Ezeknek a felépítése egyszerűbb, mint egy számlálónak. Van két bemenet, egy engedélyező jel, illetve egy jel, aminek nézzük a felfutó vagy lefutó élét, kimenete pedig jel am pontosan egy ciklusig magas jelváltáskor. Ezek használata sokban megkönnyítik a PLC kód írását.

A következő funkcióblokk kevesebb funkciót lát el, alapvetően csak számátváltásra szolgál. Mivel a kijelzőn jól mutat az, hogy a számláló folyamatosan számol vissza, nekem viszont másodpercben vannak meg az adatok el kellett végezni átalakítást óra-perc-másodperc formátumba. Ez az átváltás azonban ennél nehezebb, ugyanis, a PLC-ben többféle adattípus található és szorzás nem végezhető minden adattípussal. Ezért a szorzás összeadás után a DINT típusú változót átalakítjuk DINT típussá majd a DINT típust DWORD típussá. Utóbbi átalakításra viszont a HMI megjelenítés és a további használat miatt volt szükség. Ebből az átalakításból egy példasor így néz ki:

```
Product10_Time := DINT_TO_DWORD( REAL_TO_DINT(( ProductHour10
* 3600.0 )   +   (ProductMinute10 * 60.0)));
```

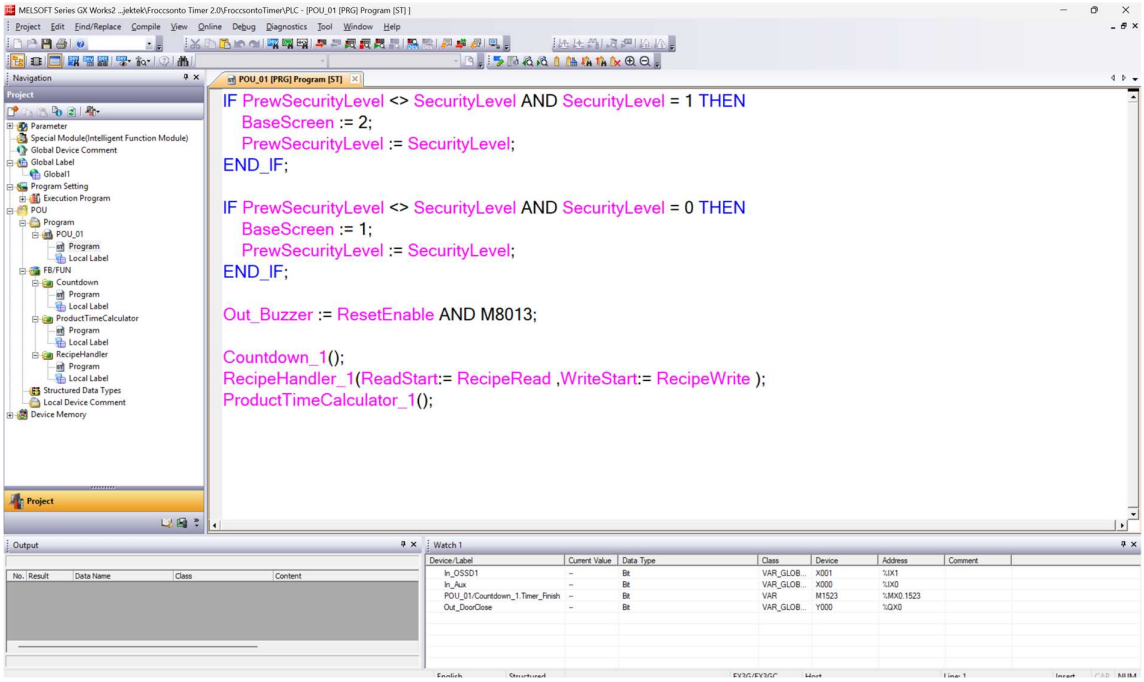
A kód számomra legösszetettebb része a következő volt: a szárítási időt HMI-n keresztül kell tudni állítani minden Admin típusú felhasználónak minden egyes termékhez. Viszont ezt a PLC-ben kell használni és nem törlődhet minden egyes kikapcsolásnál. A megoldás kidolgozásában kollégáim nyújtottak segítséget; a projekt során először alkalmaztam ezt az új módszert. Ez az úgynevezett receptkezelés, melyben létrehozunk egy listát a változókról és tartoznak hozzá értékek, a PLC ezt minden induláskor beolvassa és beírja az értékeket a saját változóiba. A receptet a HMI programjában kell létrehozni, azt részletesen ott fogom bemutatni. Viszont a recept szerkesztését PLC-ből kell programozni. Ehhez négyfunkció tartozik: írás, olvasás és megnyitás és mentés. Olvasásra és írásra létrehoztam két kicsi állapotgépet, mely vezérli ennek a folyamatát az indító jelre.

A funkcióblokkok létrehozását követően a main programot hoztam létre. Ennek a megírása ebben az esetben rövid ideig tartott, nem volt sok eszköz, funkcióblokk. Példányosítottam a funkcióblokkokat, majd a felhasználóváltást egészítettem ki, mert azt HMI-ből nem lehetett

teljesen megírni, ugyanis a régebbi HMI-ben a gombokra bizonyos funkciók mellé nem lehetett másik funkciót társítani már. Ilyen a kijelentkezés, ha az aktív akkor képernyőváltást már nem tudok társítani a gomb megnyomásához.

A felhasználóváltáshoz kötöttem két képernyőváltást. Valamint megírtam a csipogó programját, ami annyi, hogy ha lejár a program elkezd csipogni a beállított frekvenciával. Az elkészült program a három funkcióblokkal így néz ki. Bal oldalt látható a tagolás – main és alatta a funkcióblokkok a saját változóikkal és főprogramukkal.

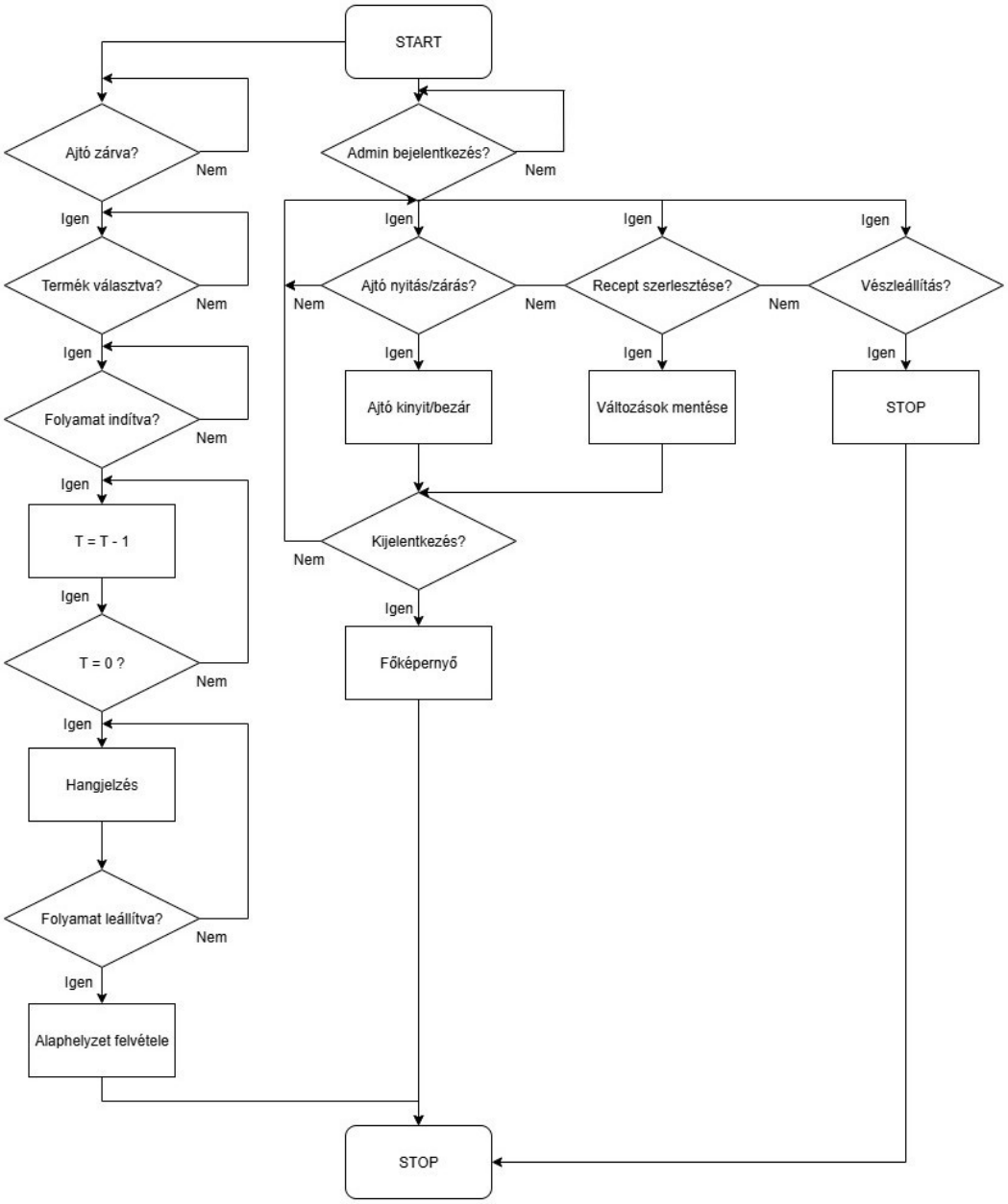
13.ábra: Főprogram



A következő ábrán bemutatom a programot folyamatábrán, előtte azonban összefoglalom, a programot.

Ahogyan az ábrán látható a kód egyaránt tartalmaz biztonsági, kezelő és adminisztratív funkciókat is. Az időzítő a start gomb megnyomásával indul, ha minden bemenő feltétel teljesül (ajtózár állapota, termék kiválasztása). Ha minden teljesült az ajtózár bezár és elindul a visszaszámlálás. Külön ágban szerepel, hiszen bármikor elérhető az Admin beállítások, ahol a receptet lehet szerkeszteni, kinyitni, bezárni az ajtót és megállítani a folyamatot bármikor.

14.ábra: Program folyamatábrája



4.1.2. HMI szerkesztés

A PLC program elkészítését követően a GT Designer 3 szoftverben az előző lépésekhez hasonlóan felmértem a szükséges beállításokat. Ebben az esetben nem változókat hoztam létre, hanem képernyőket. Ehhez a projekthez öt képernyőre volt szükségem:

- Base: ez a főképernyő, itt lehet kiválasztani a megfelelő terméket
- Timer: itt érhető el az Admin beállítás, ajtó nyitása, folyamat megállítása
- Run: ezen a kijelzőn látható, hogy hol tart a szárítási folyamat, visszaszámláló és start gomb található rajta
- Product1/Product2: ezeken az oldalakon látható a termékek neve és a szárítási idő. Itt lehet szerkeszteni a receptet, melynek van két dedikált gombja is, a mentés és a megnyitás.

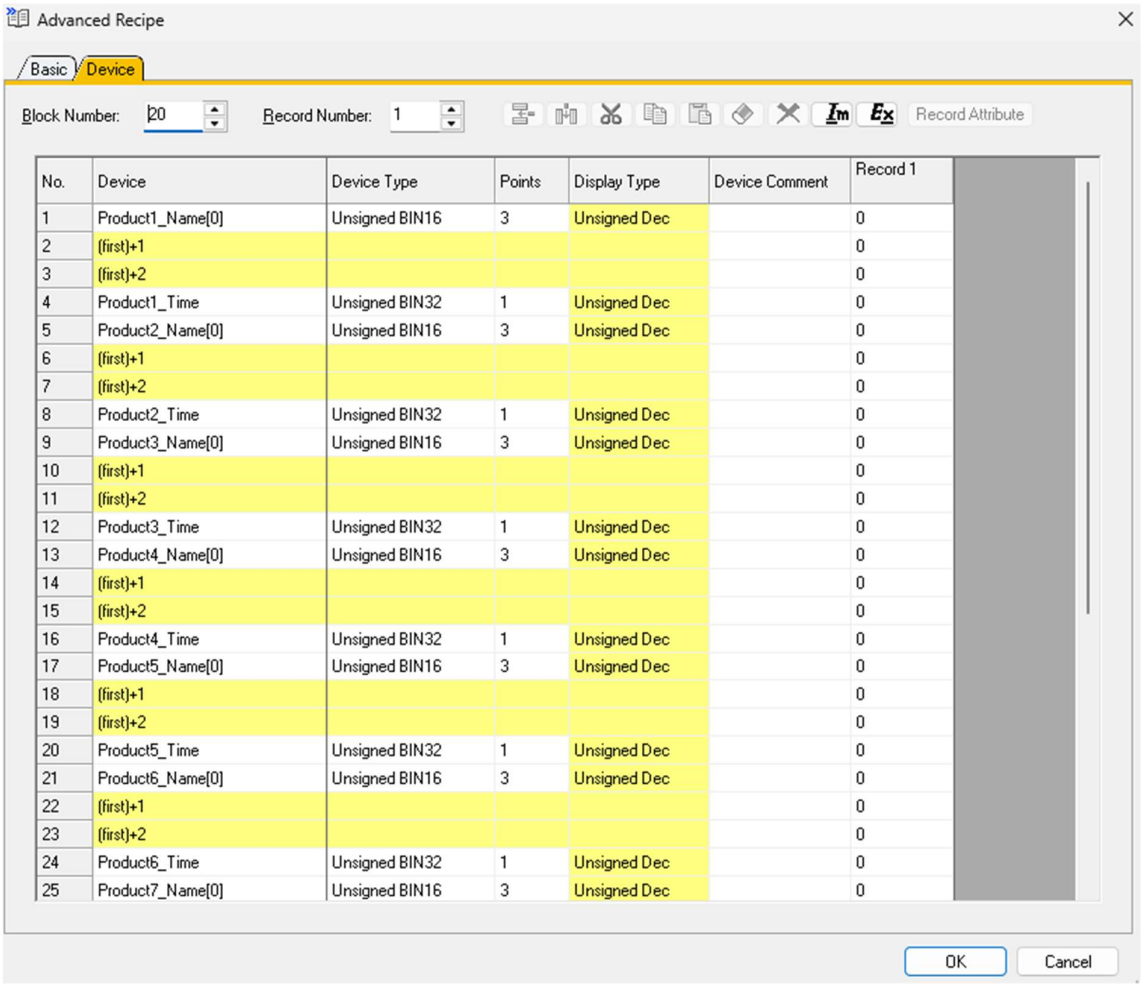
Ezen kívül, van egy minden képernyőn megjelenő sáv alul, ahol a navigációhoz szükséges gombok jelennek meg. Ilyen a bejelentkezés, kilépés beállítások.

Néhány gomb csak bizonyos szintű felhasználóknak érhető el. Erre azért van szükség, hogy a beállításokat ne tudja bárki szerkeszteni. Ennek megoldásaként hoztam létre a felhasználókezelést, aminek a neve ebben az esetben ’Security Level’. Ha a felhasználó megnyomja mondjuk a bejelentkezés gombot, felugrik egy ablak, ami kér egy jelszól. Ha a jelszó helyes tovább engedi a képernyőre. Ha visszalép a felhasználó a főképernyőre, vagy megnyomja a kijelentkezés gombot akkor automatikusan visszaáll a felhasználó szintje is. Ennél a projektnél csak két felhasználó van 0 – operátor, 1 – adminisztrátor, de lényegesen több felhasználót lehet létrehozni.

Másik funkció, amit a HMI-ben kellett leprogramozni, az a fentebb említett receptkezelés. A receptet létre kell hozni, majd a megfelelő cellákat elnevezni és az ahhoz tartozó oszlopok tartalmazzák majd termékenként az információkat, amik megjelennek a képernyőn, meg amiket használ a PLC kód is. Ahogy a képen is látható minden egyes terméknél üresen van hagyva két

hely, erre azért van szükség, hogy a termék neve biztosan elférjen a változóban. A gombok feliratát praktikus kommentként megadni. Ennek a módszernek akkor van igazán előnye, ha a projekt többnyelvű. Így egyszerűbb meg lehet oldani (mintha annyi kijelzőt készítenénk, ahány nyelven szükséges a program) egy nyelvválasztó gombbal, hogy az adott gombon hányas számú

15.ábra: Recept



No.	Device	Device Type	Points	Display Type	Device Comment	Record 1
1	Product1_Name[0]	Unsigned BIN16	3	Unsigned Dec		0
2	(first)+1					0
3	(first)+2					0
4	Product1_Time	Unsigned BIN32	1	Unsigned Dec		0
5	Product2_Name[0]	Unsigned BIN16	3	Unsigned Dec		0
6	(first)+1					0
7	(first)+2					0
8	Product2_Time	Unsigned BIN32	1	Unsigned Dec		0
9	Product3_Name[0]	Unsigned BIN16	3	Unsigned Dec		0
10	(first)+1					0
11	(first)+2					0
12	Product3_Time	Unsigned BIN32	1	Unsigned Dec		0
13	Product4_Name[0]	Unsigned BIN16	3	Unsigned Dec		0
14	(first)+1					0
15	(first)+2					0
16	Product4_Time	Unsigned BIN32	1	Unsigned Dec		0
17	Product5_Name[0]	Unsigned BIN16	3	Unsigned Dec		0
18	(first)+1					0
19	(first)+2					0
20	Product5_Time	Unsigned BIN32	1	Unsigned Dec		0
21	Product6_Name[0]	Unsigned BIN16	3	Unsigned Dec		0
22	(first)+1					0
23	(first)+2					0
24	Product6_Time	Unsigned BIN32	1	Unsigned Dec		0
25	Product7_Name[0]	Unsigned BIN16	3	Unsigned Dec		0

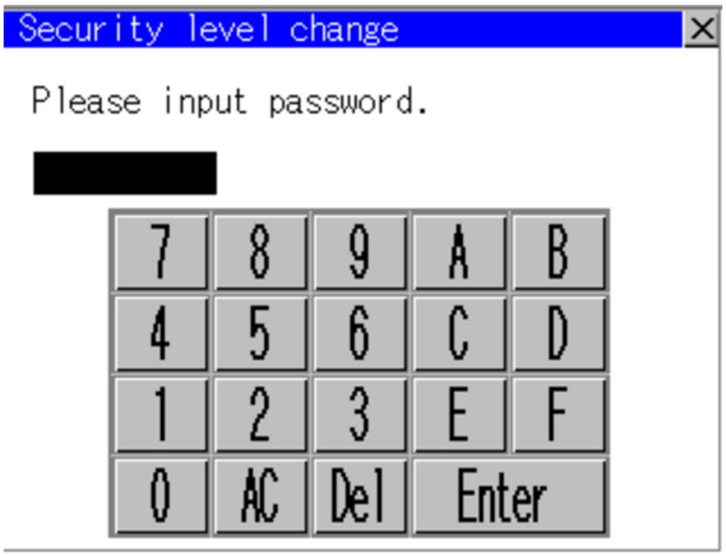
oszlopban található nevek jelenjenek meg. Jelenleg csak magyar nyelv érhető el, viszont gyorsít a programozási folyamaton, hiszen, ha több azonos nevű gomb van a nevet elég csak egyszer beállítani, majd hozzárendelni a megfelelő kommentet. Ez látható a 9. ábrán.

16.ábra: Komment

Column No.						High Quality Font					
Comment No.	1	2	Text	Invert	Blink	12dot Gothic	12dot Mincho	16dot Gothic	16dot Mincho	Style	Solid
1	100 000			No	None					Regular	
2	200 000			No	None					Regular	
3	300 000			No	None					Regular	
4	400 000			No	None					Regular	
5	500 000			No	None					Regular	
6	600 000			No	None					Regular	
7	700 000			No	None					Regular	
8	800 000			No	None					Regular	
9	900 000			No	None					Regular	
10	100 000			No	None					Regular	
11	Bejelentkezés			No	None					Regular	
12	Mentés			No	None					Regular	
13	Start			No	None					Regular	
14	Stop			No	None					Regular	
15	Reset			No	None					Regular	
16	Megnyitás			No	None					Regular	

Az alábbi fejezetben látható a HMI kezelőfelülete és egy rövid leírás, hogy mely gombokhoz milyen funkciók tartoznak, hogyan lehet navigálni az oldalak közt. Elsődleges cél az volt, hogy felhasználóbarát, könnyen kezelhető felület legyen, amin minden elvárt funkció látható és használható. Illetve, fontos volt meghatározni, hogy bizonyos szintű felhasználók mit állíthatnak, mihez van hozzáférésük. Amikor rányomnak a 'Beállítások' nevű gombra, az alábbi képernyő jelenik meg, ahol be kell írni a megfejelő jelszót és akkor megtörténik a felhasználóváltás és ezzel egyidőben a HMI-n megjelenik a másik képernyő is. Ez a felület látható a 10. ábrán.

17.ábra: Jelszóbeviteli felület

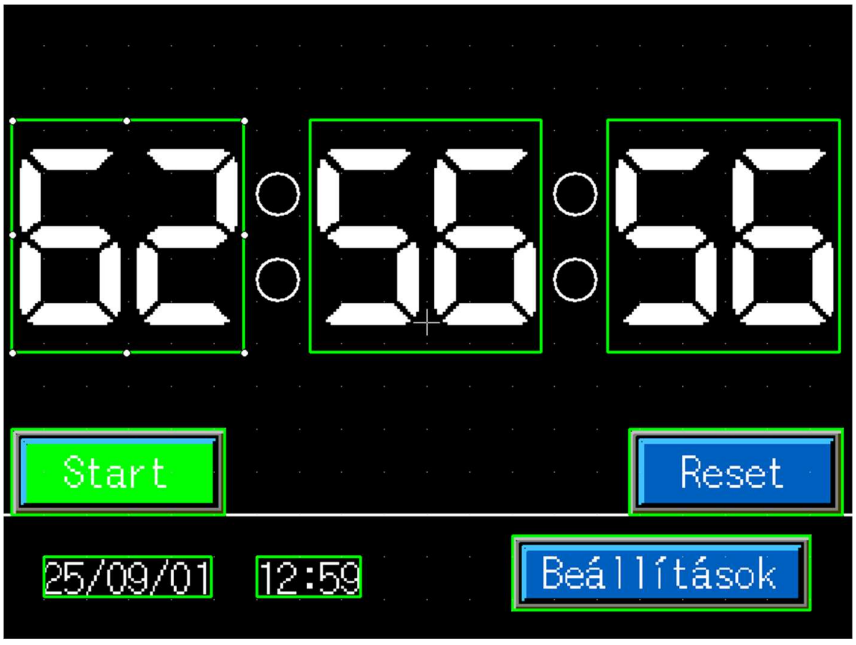


Amikor egy felhasználó rányom egy termékre a ’Base’-ről átkerül a ’Run’ nevű képernyőre.

18.ábra: Főképernyő

Ha ott megnyomja a start gombot, elindul a folyamat, ezzel együtt pedig a visszaszámlálás. Az ajtót ilyenkor már nem lehet kinyitni, bekapcsol a mágneszár. ’Reset’ nevű gombot a ’Run’ képernyőn csak akkor tudja az operátor megnyomni, amikor a folyamat lejárt és már sípol a csipogó. A csipogó egészen addig sípol, amíg a ’Reset’ gombot meg nem nyomják. A

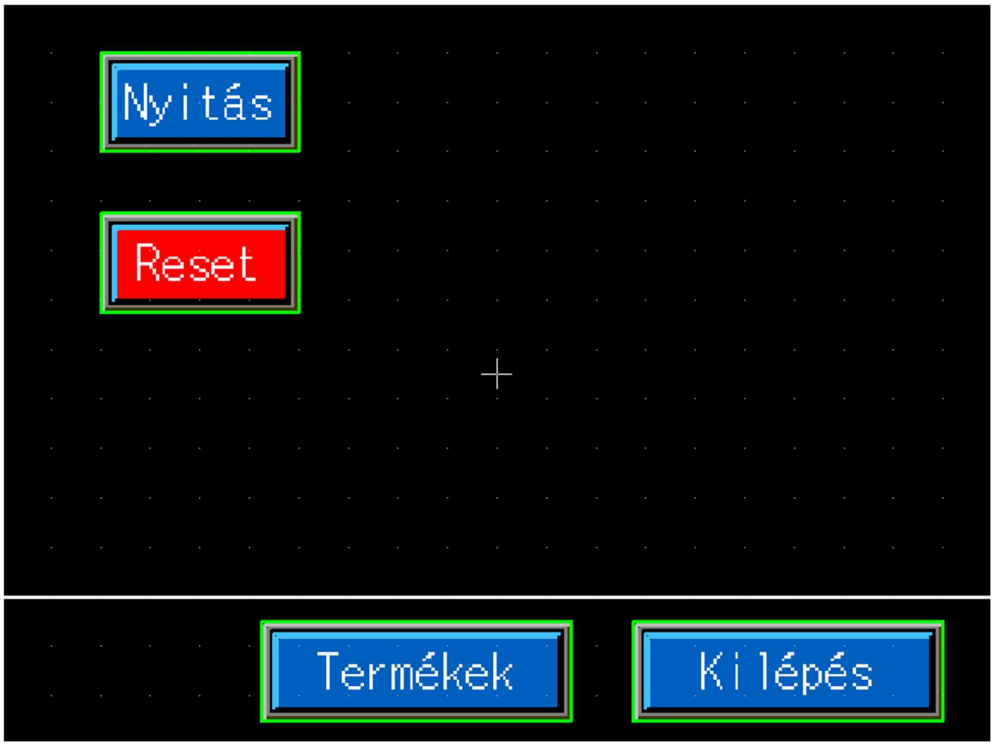
19.ábra: számláló



beállítások gombbal érhető el a ’Timer’ nevű oldal, ami csak Adminisztrátori felhasználóknak van.

Ezen a képernyőn a piros ’Reset’ gombbal azonnal megáll a folyamat és nullázódik a visszaszámlálás. A nyitás/zárás gombbal pedig a mágneses ajtózárat lehet kinyitni, illetve bezárni. Ezen a képernyőn található még a ’Termékek’ gomb, mely a ’Product1’ oldalra visz, ahol a receptet lehet szerkeszteni. Itt először meg kell nyitni a receptet a ’Megnyitás’ gombbal, majd szerkeszteni a kívánt termék nevét és/vagy értékét majd a ’Mentés’ gombbal lehet elmenteni a receptet. A ’Kilépés’ gomb megnyomásával visszatérünk a főképernyőre és visszaállítja a felhasználót az alap, operátori szintre.

20.ábra: admin felület



Szent István Campus, Gödöllő
Cím: 2100 Gödöllő, Páter Károly utca 1.
Tel.: +36-28/522-000
Honlap: https://godollo.uni-mate.hu

21.ábra: termék szerkesztése 1.

22.ábra: termék szerkesztése 2.

Szent István Campus, Gödöllő
Cím: 2100 Gödöllő, Páter Károly utca 1.
Tel.: +36-28/522-000
Honlap: https://godollo.uni-mate.hu

4.2. Berendezéshez kapcsolódó számítások

Az alkatrészek megrendelése előtt elvégeztem néhány számítást. Első sorban kiszámoltam, hogy mekkora áramot vesz fel a rendszer és mekkora teljesítményre van szükség a biztonságos működéshez. A rendszerben az alábbi fogyasztókat vettem figyelembe:

- Mitsubishi FX3G-24M PLC
- Mitsubishi GT1455-QTBD(E) HMI
- E RMQ-Titan IP40
- Keyence GS-M5 biztonsági ajtózár

3.táblázat: alkatrészek teljesítménye

Név	Teljesítmény [W]	Áramfelvétel [mA]
PLC	21	-
HMI	8,4	-
Ajtózár	6	-
Buzzer	-	18-30

A buzzer-nél az adatlapon nem volt megadva teljesítmény, így azt ki kellett számolni az alábbi képlet felhasználásával:

(1) $$P = U * I$$

$$P = 24 * 0{,}3 = 7{,}2$$

Ahol:

P: teljesítmény [W],
U: feszültség [V],
I: áramerősség [A].

A képletbe a maximum fogyasztással számolva és behelyettesítve megkapjuk, hogy a buzzer teljesítménye 7,2 W.

Miután kiszámoltam a komponensek áramfelvételét és teljesítményét, választottam hozzá egy megfelelő tápegységet. Számításaim alapján a rendszer teljesítménye:

(2) $$\sum P = P1 + P2 + P3 + P4$$

$$\sum P = 21 + 8{,}4 + 6 + 7{,}2 = 42{,}6$$

Tápegység kiválasztásánál fontos úgy méretezni, hogy a rendszer a tápegység nagyjából 60-80%-át használja [35], így biztosak lehetünk benne, hogy a tápegység megfelelő áramot és teljesítményt tud biztosítani. Ezek alapján az alábbi tápegységet választottam: MeanWell MDR_60-24, mely 60W-os. Ebből a rendszer 71%-ot használ, azonban a kínálatból ez állt a legközelebb a számításaimhoz és még így is van durván 30% tartalék a tápegységben.

Továbbá, ki tudjuk számolni a rendszer áramfelvételét is erre azért van szükség, hogy később meg tudom határozni, hogy a kábel megfelelő keresztmetszetű-e. 24 V-os egyenfeszültségen a teljesítményből kiszámolom a komponensek áramerősségét és összeadjuk őket az alábbi módon:

(3) $$\sum I = \frac{P_1}{U_1} + \frac{P_2}{U_2} + \frac{P}{U_3} + I_4$$

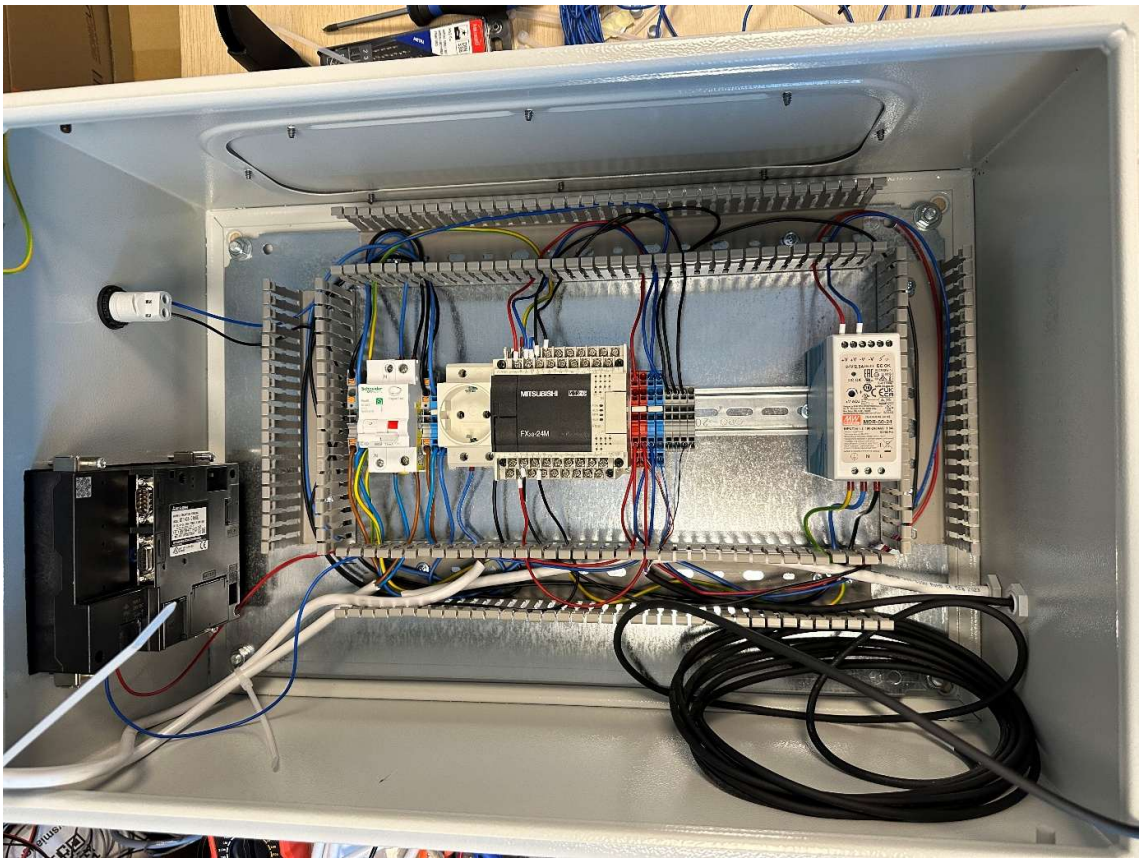
$$\sum I = \frac{21}{24} + \frac{8{,}4}{24} + \frac{6}{24} + 0{,}3 = 1{,}775$$

Tehát a rendszer maximum 1,775 A áramot vesz fel. Ez alapján ki lehetne számolni egyenletek, illetve táblázatok alapján a kábelvastagságot. Azonban cégen belül konvenció szerint a magasfeszültségű váltakozóáramú részt 1,5mm^2 keresztmetszetű, az egyenáramú kisfeszültségű részt pedig 0,75mm^2 keresztmetszetű vezetékkel csináltam.

Szent István Campus, Gödöllő
Cím: 2100 Gödöllő, Páter Károly utca 1.
Tel.: +36-28/522-000
Honlap: https://godollo.uni-mate.hu

Az összeszerelt gép látható a 16.ábrán.

23.ábra: Kábelezés



Szent István Campus, Gödöllő
Cím: 2100 Gödöllő, Páter Károly utca 1.
Tel.: +36-28/522-000
Honlap: https://godollo.uni-mate.hu

5. Gazdasági számítás

Ebben a fejezetben a berendezés gazdasági vonzatát tárgyalom. A különböző alkatrészek árát és a Magyar Mérnöki Kamara aktuális ár táblázata segítségével határozom meg a gép árát. Megtérülést viszonylag bonyolult lenne képlettel számolni egy ilyen berendezésnél, hiszen közvetlenül nem termel hasznot az üzemeltető cég számára. Azonban, ha emberi hiba történne és nem a gép időzítene és szárítaná a műanyagot az anyagi kár mértéke nagyon magas lenne. Ezért mondhatjuk, hogy nagyjából kettő, három emberi tévesztés alatt megtérül. Egy cég számára, ez a befektetés egyértelműen megérni, ha hosszú távon tervezik használni a gépet. Az alábbi táblázatban az alkatrészek kereskedelmi ára látható:

4.táblázat: Árjegyzék

Megnevezés	Ár [Ft, bruttó]
MeanWell MDR-60-24 tápegység	8.730 (www.tme.hu)
Mitsubishi Fx3g-24m PLC	201.127 Ft (hu.rs-online.com)
Schneider áramvédős kismegszakító	15.959 Ft (www.se.com)
Mitsubishi GOT1000 HMI	197.316 Ft (hu.wiautomation.com)
EATON RMQ-Titan IP40 Buzzer	20.523 Ft (hu.rs-online.com)
Keyence GS-M51P	115.000 Ft (www.keyence.hu)
Schneider kapcsolószekrény szerelőlappal	57.100 Ft (www.se.com)
Egyéb költségek (vezeték, sorkapcsok érvéghüvely, tömbszelence, kábelcsatorna)	15.000 Ft

Ezen alkatrészek összesen 630.755 Forint. Ehhez hozzáadódik még a munkadíj, mely több részre oszlik. Tervezés, programozás szerelés. Az alábbi táblázatban látható ezen munkafolyamatokra szánt órák, illetve a mérnöki kamara által szolgáltatott óradíjak kezdő mérnökre vonatkoztatva. Ez az érték 2025-ben 123.000 Ft, ami azt jelenti, hogy 15.375Ft/óra. Ez alapján állítottam össze a táblázatot.

Szent István Campus, Gödöllő
Cím: 2100 Gödöllő, Páter Károly utca 1.
Tel.: +36-28/522-000
Honlap: https://godollo.uni-mate.hu

5.táblázat: Munkadíjak

Név	Óra	Összesen [Ft]
Tervezés	4	61.500
Programozás	24	369.000
Szerelés	16	246.000
Telepítés	8	123.000

Ezek összesen 799.500 Ft. Ez azt jelenti, hogy az egész berendezés anyag és munkaköltsége összesen 1.430.255 Ft nettó.

6. Összefoglalás

Az elkészült berendezés elsődleges célja a tizenkét féle műanyag szárítása, melyek mind más-más száradási idővel rendelkeznek. Mivel a száradás magas hőmérsékleten történik, biztonsági okokból szükség van egy mágneses biztonsági ajtózárra is. A gépet egy vezérlőegység irányít, melyen az előre megírt program fut. A száradási idő lejártát egy hangjelzés követi, aminek resetelésével old a biztonsági ajtózár és nyitható a kamra.

Tervezés során fő szempont volt, hogy megfeleljen a megrendelői elvárásoknak, többek közt, legyen a programban felhasználókezelés, ajtózár, időzítő, mely termékenként testre szabható és módosítható utólag a HMI-ből, legyen hangjelzés a folyamat végén, a grafikus felület pedig legyen felhasználóbarát, informatív, áttekinthető és gyorsan tanulható.

A szakdolgozat elsődleges célja volt, hogy bemutassam lépésről lépésre, hogyan készül el egy ilyen berendezés. Első lépésként megírtam a vezérlőprogramot, mely a PLC-n fut. Ez irányítja a berendezést, a PLC kezeli a ki és bemenő jeleket, ilyenek az ajtózár, hangjelzés. Ez után megszerkesztettem a HMI-t és összekapcsoltam a PLC-vel.

Miután szoftveresen befejeztem a munkát, elkezdtem összeépíteni fizikailag a gépet. Méreteztem a tápegységet, kiszámoltam az össz áramfelvételt és ennek segítségével vezetéket méreteztem. Ezután összeépítettem a komponensekből a berendezés és elvégeztem a kábelezést.

Miután minden szükséges elem a helyére került teszteltem a berendezésen futó programot, és kijavítottam a felmerülő hibákat.

A tesztelés és javítás után az általam tervezett rendszer megfelel a megrendelő elvárásainak, biztosítja a hibamentes működést és az operátorok és kezelő személyek megfelelő biztonságát.

Szent István Campus, Gödöllő
Cím: 2100 Gödöllő, Páter Károly utca 1.
Tel.: +36-28/522-000
Honlap: https://godollo.uni-mate.hu

7. Summary

The primary purpose of the completed equipment is to dry twelve different types of plastic, each with its own drying time. Since the drying process takes place at elevated temperatures, a magnetic safety door lock is also needed for safety reasons. The machine is controlled by a programmable logical controller, which runs a pre-programmed sequence.

During the design process, the main goal was to meet the client's expectations. This included features such as user management in the program, a door lock, a timer that can be customized and adjusted for each product, an audible signal at the end of the process, and a graphical interface that is user-friendly, clear, and easy to learn.

The main goal of my thesis was to show, step by step, how a machine like this is built. As a first step, I wrote the control program that runs on the PLC. This program controls the entire system, the PLC manages the input and output signals, such as the door lock and the sound alert. After that, I designed the HMI interface and connected it to the PLC.

Once the software part was finished, I started building the machine itself. I calculated the power supply capacity, estimated the total current consumption, and based on that, I selected the proper wiring. Then I assembled the components and completed the wiring of the equipment.

After everything was in place, I tested the program running on the machine and corrected any errors that appeared during the process.

The system I designed and built meets the client's requirements, ensures reliable operation, and provides proper safety for operators and for other people.

8. Nyilatkozat

MATE Szervezeti és Működési Szabályzat
III. Hallgatói Követelményrendszer
III.1. Tanulmányi és Vizsgaszabályzat
6.13. sz. függeléke: A MATE egységes szakdolgozat / diplomadolgozat / záródolgozat / portfólió készítési útmutatója
7. sz. melléklete: Műszaki Intézet külső konzulensi nyilatkozat

KÜLSŐ KONZULENSI NYILATKOZAT

Bognár Botond (név) (hallgató Neptun azonosítója: YWIKY1)

külső konzulenseként nyilatkozom arról, hogy a hallgató az előre egyeztetett konzultációkon rendszeresen megjelent.

Kelt: 2025. év 10. hó 22. nap

KULCSÁR DÁVID
külső konzulens

ElectroZen Kft.
1117 Budapest
Baranyai u. 27. 3.em. 3.ajtó
Adószám: 32625956-2-43
Cégjegyzékszám: 01-09-434171

MAGYAR AGRÁR- ÉS ÉLETTUDOMÁNYI EGYETEM

Szent István Campus, Gödöllő
Cím: 2100 Gödöllő, Páter Károly utca 1.
Tel.: +36-28/522-000
Honlap: https://godollo.uni-mate.hu

MATE Szervezeti és Működési Szabályzat
III. Hallgatói Követelményrendszer
III.1. Tanulmányi és Vizsgaszabályzat
6.13. sz. függeléke: A MATE egységes szakdolgozat / diplomadolgozat / záródolgozat / portfólió készítési útmutatója
4.2. sz. melléklete: Nyilatkozat a záródolgozat/szakdolgozat/diplomadolgozat/portfólió nyilvános hozzáféréséről és eredetiségéről (módosítva: 2025. október 16.)

NYILATKOZAT

szakdolgozat nyilvános hozzáféréséről és eredetiségéről

A hallgató neve:	Bognár Botond
A Hallgató Neptun kódja:	YWIKY1
A dolgozat címe:	Célgép PLC programja
A megjelenés éve:	2025
A konzulens intézetének neve:	Műszaki Intézet
A konzulens tanszékének a neve:	Mechatronika Tanszék

Kijelentem, hogy az általam benyújtott záródolgozat/szakdolgozat/diplomadolgozat/portfólió[1] egyéni, eredeti jellegű, saját szellemi alkotásom. Azon részeket, melyeket más szerzők munkájából vettem át, egyértelműen megjelöltem, és az irodalomjegyzékben szerepeltettem. Továbbá kijelentem, hogy a dolgozat elkészítése során alkalmazott mesterséges intelligencia-eszközök (pl. szöveggenerálás, nyelvi javítás, fordítás, adatelemzés) használata nem helyettesítette a saját kutatási és alkotói munkámat, azok alkalmazását a források között vagy a módszertani részben feltüntettem, és a szakmai-etikai elvárásoknak megfelelően jártam el.

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

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

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

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

- nem titkosított dolgozat a védést követően
- titkosításra engedélyezett dolgozat a benyújtásától számított 5 év eltelte után

nyilvánosan elérhető és kereshető lesz az Egyetem könyvtári repozitori rendszerében.

Kelt: 2025 év 10 hó 25 nap

Hallgató aláírása

MAGYAR AGRÁR- ÉS ÉLETTUDOMÁNYI EGYETEM

Szent István Campus, Gödöllő
Cím: 2100 Gödöllő, Páter Károly utca 1.
Tel.: +36-28/522-000
Honlap: https://godollo.uni-mate.hu

NYILATKOZAT

Bognár Botond (név) (hallgató Neptun azonosítója: YWIKY1) konzulenseként nyilatkozom arról, hogy a záródolgozatot/szakdolgozatot/diplomadolgozatot/portfóliót[1] áttekintettem, a hallgatót az irodalmi források korrekt kezelésének követelményeiről, jogi és etikai szabályairól tájékoztattam.

A szakdolgozatot a záróvizsgán történő védésre **javaslom / nem javaslom**.

A dolgozat állam- vagy szolgálati titkot tartalmaz: igen nem

Kelt: Gödöllő, 2025 év 10 hó 28 nap

belső konzulens

Szent István Campus, Gödöllő
Cím: 2100 Gödöllő, Páter Károly utca 1.
Tel.: +36-28/522-000
Honlap: https://godollo.uni-mate.hu

Hallgatók, doktoranduszok nyilatkozata mesterséges intelligencia (MI) alkalmazásáról

1. Általános adatok

Hallgató neve:	**Bognár Botond**
Neptun-kódja:	**YWIKY1**
Képzési szint (a megfelelőt jelölje X-szel):	☒ BSc/BA ☐ MSc/MA ☐ Doktori (PhD) ☐ Egyéb: ……
Tantárgy neve/kódja*:	**Szakdolgozat készítés**
A munka címe:	**Célgép PLC programja**

* doktori értekezés esetén nem kitöltendő

2. Nyilatkozat az MI használatáról

Alulírott, etikai felelősségem teljes tudatában az alábbi nyilatkozatot teszem:

(Kérjük, válasszon egyet az alábbi lehetőségek közül!)

☐ A) Nem alkalmaztam mesterséges intelligencia rendszert vagy szolgáltatást.

(Amennyiben ezt jelölte, a további táblázatok kitöltése nem szükséges.)

☒ B) Alkalmaztam mesterséges intelligencia rendszert vagy szolgáltatást.

(Kérjük, töltse ki a vonatkozó táblázatokat!)

3. A mesterséges intelligencia használatának részletezése

I. TÁBLÁZAT: Asszisztensi vagy kisebb mértékű felhasználás (pl. fordítás, nyelvi korrektúra, ötletelés stb.)

(Ezen felhasználások esetében a konkrét promptok és válaszok csatolása nem szükséges.)

A felhasználás célja	Alkalmazott MI-eszköz neve és verziója	Érintett rész (ha nem a szöveg egészére vonatkozik)
Tartalmi tervezet készítése	ChatGPT 4.0	

II. TÁBLÁZAT: Jelentős tartalmi hozzájárulás (pl. egy teljes ábra vagy egy hosszabb szövegrész generálása)

*(Ezekben az esetekben a felhasznált kulcsfontosságú promptok és az MI által adott nyers válaszok dokumentálása és a munka **mellékletében való csatolása szükséges**.)*

A felhasználás célja	Alkalmazott MI-eszköz neve, verziója, elérhetősége	Az érintett fejezet / ábra / táblázat pontos sorszáma	A prompt-naplót tartalmazó melléklet bejegyzésének sorszáma

Szent István Campus, Gödöllő
Cím: 2100 Gödöllő, Páter Károly utca 1.
Tel.: +36-28/522-000
Honlap: https://godollo.uni-mate.hu

3/A. Oktató által előírt kiegészítő szabályok (ha vannak)

Amennyiben az adott tantárgy oktatója vagy témavezetője az MI-eszközök használatára vonatkozóan külön szabályokat vagy elvárásokat határozott meg, kérjük, az alábbi mezőben foglalja össze ezeket:

Pl. az MI használatának tilalma bizonyos feladattípusokra; csak konkrét eszköz használata engedélyezett; eltérő hivatkozási elvárások; dokumentációs forma stb.

Oktató vagy témavezető által előírt szabályok:
A MATE-K/13-12/2025 iktatószámú 11/2025 (VIII. 29.) számú rektori utasítás: a MESTERSÉGES INTELLIGENCIA HASZNÁLATÁRÓL SZÓLÓ SZABÁLYZAT hatályba lépése előtt 2025. szeptember 1. előtt az MI használata engedélyezett.

4. Minden hallgatóra vonatkozó nyilatkozat:

Kijelentem, hogy az MI által esetlegesen generált tartalmakat minden esetben kritikailag felülvizsgáltam, szerkesztettem és a munkába illesztettem. A leadott munka minden eleméért, annak eredetiségéért és tudományos helytállóságáért teljes körű felelősséget vállalok. Tudomásul veszem, hogy a Magyar Agrár- és Élettudományi Egyetem a benyújtott munkát mesterséges intelligencia detektorral ellenőrizheti, és eljárást kezdeményezhet, amennyiben a nyilatkozatom valótlan vagy hiányos.

Kelt: Gödöllő 2025. .....10..... hó ..28.. nap

Hallgató aláírása | **Konzulens/Témavezető aláírása**

9. Irodalomjegyzék

[1] C. B. J. M. Mark Walker, „The PLC: a logical development," *Measurement + Control,* %1. számVol 43/9, p. Vol 43/9, 2010.

[2] B. Attila, Mechatronika alapjai, 2014.

[3] "Beckhoff.com," Beckhoff, 2025. [Online]. Available: https://www.beckhoff.com/en-en/products/ipc/software-and-tools/operating-systems/. [Accessed 30 09 2025].

[4] D. N. József, Irányítástechnika, Nyugat-magyarországi Egyetem, 2012.

[5] A. Beckmann, „AUTOMATED VERIFICATION ENVIRONMENT FOR TWINCAT PLC PROGRAMS," in *European XFEL*, Hamburg, Germany, 2013.

[6] A. Ahmed, „Automatic Control of Electrical overhead Smart Trolley Crane AEOSTC Based Programmable Logic Controller (PLC)," American Journal of Engineering Research (AJER), 2017.

[7] M. Garrick, „Machine Safety Design: Safety Relays Versus a Single Safety Controller," Phoenix Contact, Harrisburg, PA 17111-0100 , 2010.

[8] I. Siemens Industry, „Evolving from Safety Relays to Safety PLCs," Siemens Industry, Inc, U.S.A, 2018.

[9] Sick, „Sick," Sick, 20 07 2022. [Online]. Available: https://www.sick.com/us/en/what-are-performance-levels/w/blog-safety-standard-performance-levels. [Hozzáférés dátuma: 05 07 2025].

[10] „Sick connect," Sick, 03 11 2025. [Online]. Available: https://sickconnect.com/safety-performance-levels/.

[11] Encoder Products Company, "Industial Ethernet Communication Protocols," Encoder Products Company, 2019.

[12] J. A. Ildikó, „PLC programozás IEC 1131-3 szabvány szerint," PTE – PMMFK, Műszaki Informatika Tanszék .

[13] *IEC_61131-3 szabvány.*

[14] M. Chmiel, An Idea of Event-Driven Program Tasks Execution., Elsevier BV : IFAC Proceedings Volumes, 2009.

[15] Beckhoff, „beckhoff.com," BEckhoff, [Online]. Available: https://infosys.beckhoff.com/english.php?content=../content/1033/tcplclib_tc2_standard/74391563.html&id=. [Hozzáférés dátuma: 30 09 2025].

[16] M. C. Robert Czerwinski, „Hardware-Based Single-Clock-Cycle Edge Detector For a PLC Central Unit," Institute of Electronics, Silesian University of Technology, Gliwice, Poland, 2019.

[17] E. K. a. J. K. P. Papcun, Human Machine Interface in Concept of Industry 4.0, Košice, Slovakia: 2018 World Symposium on Digital Intelligence for Systems and Machines (DISA), 2018.

[18] K. Z. Pletl Szilveszter, „PLC és SCADA rendszerek," Pannon Egyetem és a Szegedi Tudományegyetem, Magyarország, 2014.

[19] AutomationDirect, „library.automationdirect," AutomationDirect, [Online]. Available: https://library.automationdirect.com/evolution-of-human-machine-interface/. [Hozzáférés dátuma: 30 09 2025].

[20] *IEC 61508 szabvány.*

[21] *ISO 13849 szabvány.*

[22] *IEC 62264 szabvány.*

[23] A. ISTVÁN, PLC ÉS SCADA-HMI RENDSZEREK I., Miskolc: AUT-INFO Kft., 2007.

[24] K. József, „A kismegszakítókról," *Villanyszerelők lapja,* p. 4., 2007.

[25] *MSZ EN 60947/2.*

[26] CD-Team, „electronicsforyou.com," 06 06 2024. [Online]. Available: https://www.electronicsforu.com/technology-trends/learn-electronics/miniature-circuit-board-mcb. [Hozzáférés dátuma: 11 03 2025].

[27] P. Gábor, Villamos kapcsolókészülékek, Budapest: Műegyetemi Kiadó, 2015.

[28] K. Zoltán, „techstorym2m," Endrich Bauelemente Vertriebs GmbH, 23 04 2018. [Online]. Available: https://www.techstorym2m.hu/dc-dc-kapcsolouzemu-feszultsegatalakitok-mukodese.html. [Hozzáférés dátuma: 11 03 2025].

[29] S. D. C. (SDC), „Electromagnetic Locks," 2í23.

[30] *IEC 60529.*

[31] „Keyence.eu," Keyence, [Online]. Available: https://www.keyence.eu/huhu/products/safety/safety-interlock-switches/gs/specs/. [Hozzáférés dátuma: 30 09 2025].

[32] „Omron.com," Omron, [Online]. Available: https://www.ia.omron.com/products/family/3736/. [Hozzáférés dátuma: 30 09 2025].

[33] D. S. Tibor, Gépészeti automatizálás, Magyarország: Edutus Főiskola, 2011.

[34] „Mitsubishi electric," Mitsubishi, 03 11 2025. [Online]. Available: https://dl.mitsubishielectric.com/dl/fa/document/manual/plc_fx/jy997d34801/jy997d34801k.pdf.

[35] „Eaton.com," Eeaton, [Online]. Available: https://www.eaton.com/in/en-us/products/controls-drives-automation-sensors/power-supplies/how-to-size-a-power-supply.html. [Hozzáférés dátuma: 30 09 2025].

[36] „Mitsubishi electric," Mitusbishi, 03 11 2025. [Online]. Available: https://mitsubishi-electric-eshop.mee.com/mee/FA_IA/en/EUR/Catalogue/PLC/PLC-Compact/Main-Unit/FX3G-24MR-ES/p/000000000000231467.

Szent István Campus, Gödöllő
Cím: 2100 Gödöllő, Páter Károly utca 1.
Tel.: +36-28/522-000
Honlap: https://godollo.uni-mate.hu

10. Mellékletek

Következő oldalakon részletezem a forráskódot, ami fut a PLC, az elektromos tervet, illetve a fejezet végére beillesztettem néhány képet az összeszerelt berendezésről.

10.1.Forráskóds

Main:

```
 IF PrewSecurityLevel <> SecurityLevel AND SecurityLevel = 1
THEN
     BaseScreen := 2;
     PrewSecurityLevel := SecurityLevel;
 END_IF;
 IF PrewSecurityLevel <> SecurityLevel AND SecurityLevel = 0
THEN
     BaseScreen := 1;
     PrewSecurityLevel := SecurityLevel;
 END_IF;
 Out_Buzzer := ResetEnable AND M8013;
 Countdown_1();
 RecipeHandler_1(ReadStart:= RecipeRead ,WriteStart:=
RecipeWrite );
 ProductTimeCalculator_1();
```

Szent István Campus, Gödöllő
Cím: 2100 Gödöllő, Páter Károly utca 1.
Tel.: +36-28/522-000
Honlap: https://godollo.uni-mate.hu

Countdown funkcióblokk:

```
Time_Left := DWORD_TO_INT(ProductTime);

IF  StartButtonPressed OR AdminOpenClose = TRUE THEN
    Out_DoorClose := TRUE;
ELSIF Timer_Finish OR AdminOpenClose = FALSE THEN
    Out_DoorClose := FALSE;
END_IF;

IF LDP(TRUE, AdminReset) THEN
    StartButtonPressed := FALSE;
    CounterReset := TRUE;
END_IF;

IF LDF(TRUE, AdminReset) THEN
    CounterReset := FALSE;
END_IF;

IF StartButtonTimer  AND In_Aux = FALSE THEN
StartButtonPressed := TRUE;
END_IF;

CounterEnableTimer := StartButtonPressed AND M8013 AND
NOT(Timer_Finish) ;

IF Timer_Finish THEN
    StartButtonPressed := FALSE;
END_IF;

IF LDP(TRUE, CounterReset) THEN
    StartButtonPressed := FALSE;
END_IF;
```

```
PLS(CounterReset OR M8002, Reset);

ResetEnable := Timer_Finish;

CTU_2(CU:= CounterEnableTimer ,RESET:= Reset ,PV:= Time_Left
,Q:= Timer_Finish ,CV:= Time_Elapsed );

TimeLeft := INT_TO_DINT( Time_Left - Time_Elapsed);
DDIV( TRUE ,TimeLeft , 3600 , HourResult );
DDIV(TRUE, HourResult[1], 60, MinuteResult);

i_Countdown_Hour := DWORD_TO_INT( HourResult[0]);
i_Coundown_Minute := DWORD_TO_INT( MinuteResult [0]);
i_Countdown_Second := DWORD_TO_INT( MinuteResult[1]);
```

Product Time Calculator funkcióblokk:

```
Product1_Time := DINT_TO_DWORD( REAL_TO_DINT(( ProductHour1 *
3600.0 )   +   (ProductMinute1 * 60.0)));
Product2_Time := DINT_TO_DWORD( REAL_TO_DINT(( ProductHour2 *
3600.0 )   +   (ProductMinute2 * 60.0)));
Product3_Time := DINT_TO_DWORD( REAL_TO_DINT(( ProductHour3 *
3600.0 )   +   (ProductMinute3 * 60.0)));
Product4_Time := DINT_TO_DWORD( REAL_TO_DINT(( ProductHour4 *
3600.0 )   +   (ProductMinute4 * 60.0)));
Product5_Time := DINT_TO_DWORD( REAL_TO_DINT(( ProductHour5 *
3600.0 )   +   (ProductMinute5 * 60.0)));
Product6_Time := DINT_TO_DWORD( REAL_TO_DINT(( ProductHour6 *
3600.0 )   +   (ProductMinute6 * 60.0)));
Product7_Time := DINT_TO_DWORD( REAL_TO_DINT(( ProductHour7 *
3600.0 )   +   (ProductMinute7 * 60.0)));
Product8_Time := DINT_TO_DWORD( REAL_TO_DINT(( ProductHour8 *
3600.0 )   +   (ProductMinute8 * 60.0)));
Product9_Time := DINT_TO_DWORD( REAL_TO_DINT(( ProductHour9 *
3600.0 )   +   (ProductMinute9 * 60.0)));
```

Szent István Campus, Gödöllő
Cím: 2100 Gödöllő, Páter Károly utca 1.
Tel.: +36-28/522-000
Honlap: https://godollo.uni-mate.hu

```
 Product10_Time := DINT_TO_DWORD( REAL_TO_DINT(( ProductHour10
* 3600.0 )   +   (ProductMinute10 * 60.0)));
```

Recipe Handler fukcióblokk:

```
RecipieNoStorageDevice := 1;
RecordNoStorageDevice := 1;

IF LDP(TRUE, ReadStart) AND State = 0 THEN
    State := 10;
END_IF;

IF LDP(TRUE, WriteStart) AND StateWrite = 0 THEN
    StateWrite := 10;
END_IF;

CASE State OF
    10:
         ReadError := FALSE;
         ReadCompleted := FALSE;
         ExternalControlDevice := 1;
         IF (ExternalNotificationDevice AND 1) = 1 THEN
              State := 20;
         END_IF;
    20:
         ExternalControlDevice := 0;
         IF ( ExternalNotificationDevice AND 1) = 0 THEN
              State := 30;
         END_IF;
    30:
         IF (ExternalNotificationDevice AND 16) = 16 THEN
              ReadCompleted := TRUE;
              State := 0;
         END_IF;
```

Szent István Campus, Gödöllő
Cím: 2100 Gödöllő, Páter Károly utca 1.
Tel.: +36-28/522-000
Honlap: https://godollo.uni-mate.hu

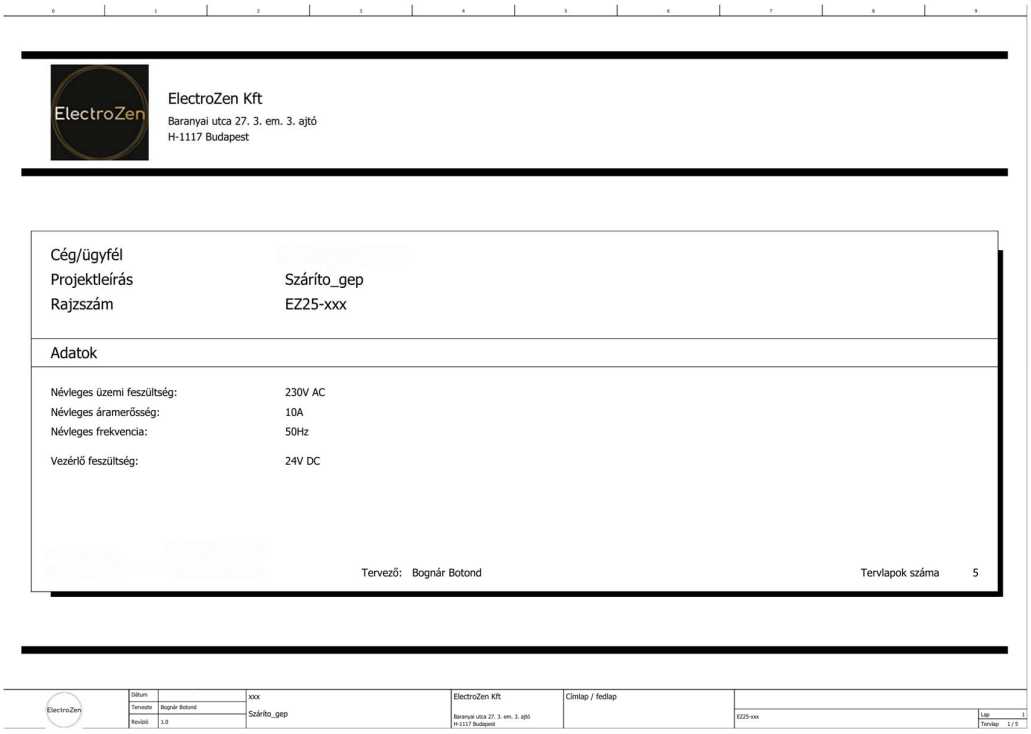
```
          IF (ExternalNotificationDevice AND 32768) = 32768
THEN
               ReadError := TRUE;
               State := 40;
          END_IF;
     40:
          ExternalControlDevice := 32768;
          IF  (ExternalNotificationDevice AND 32768) = 0 THEN
               ExternalControlDevice := 0;
               State := 0;
          END_IF;
 END_CASE;

 CASE StateWrite OF
     10:
          WriteError := FALSE;
          WriteCompleted := FALSE;
          ExternalControlDevice := 2;
          IF (ExternalNotificationDevice AND 2) = 2 THEN
               StateWrite := 20;
          END_IF;
     20:
          ExternalControlDevice := 0;
          IF (ExternalNotificationDevice AND 2) = 0 THEN
               StateWrite := 30;
          END_IF;
     30:
          IF (ExternalNotificationDevice AND 32) = 32 THEN
               WriteCompleted := TRUE;
               StateWrite := 0;
          END_IF;
          IF (ExternalNotificationDevice AND 32768) = 32768
THEN
               WriteError := TRUE;
```

Szent István Campus, Gödöllő
Cím: 2100 Gödöllő, Páter Károly utca 1.
Tel.: +36-28/522-000
Honlap: https://godollo.uni-mate.hu

```
            StateWrite := 40;
        END_IF;
    40:
        ExternalControlDevice := 32768;
        IF  (ExternalNotificationDevice AND 32768) = 0 THEN
            ExternalControlDevice := 0;
            StateWrite := 0;
        END_IF;
END_CASE;
```

10.2. Elektromos terv

24.ábra: Elektromos terv 1.oldal



ElectroZen Kft
Baranyai utca 27. 3. em. 3. ajtó
H-1117 Budapest

Cég/ügyfél	
Projektleírás	Száríto_gep
Rajzszám	EZ25-xxx

Adatok

Névleges üzemi feszültség:	230V AC
Névleges áramerősség:	10A
Névleges frekvencia:	50Hz
Vezérlő feszültség:	24V DC

Tervező: Bognár Botond

Tervlapok száma 5

ElectroZen	Dátum		xxx	ElectroZen Kft Baranyai utca 27. 3. em. 3. ajtó H-1117 Budapest	Címlap / fedlap
	Tervezte	Bognár Botond	Száríto_gep		
	Revízió	1.0			EZ25-xxx Lap 1 Tervlap 1 / 5

25.ábra: Elektromos terv 2.oldal

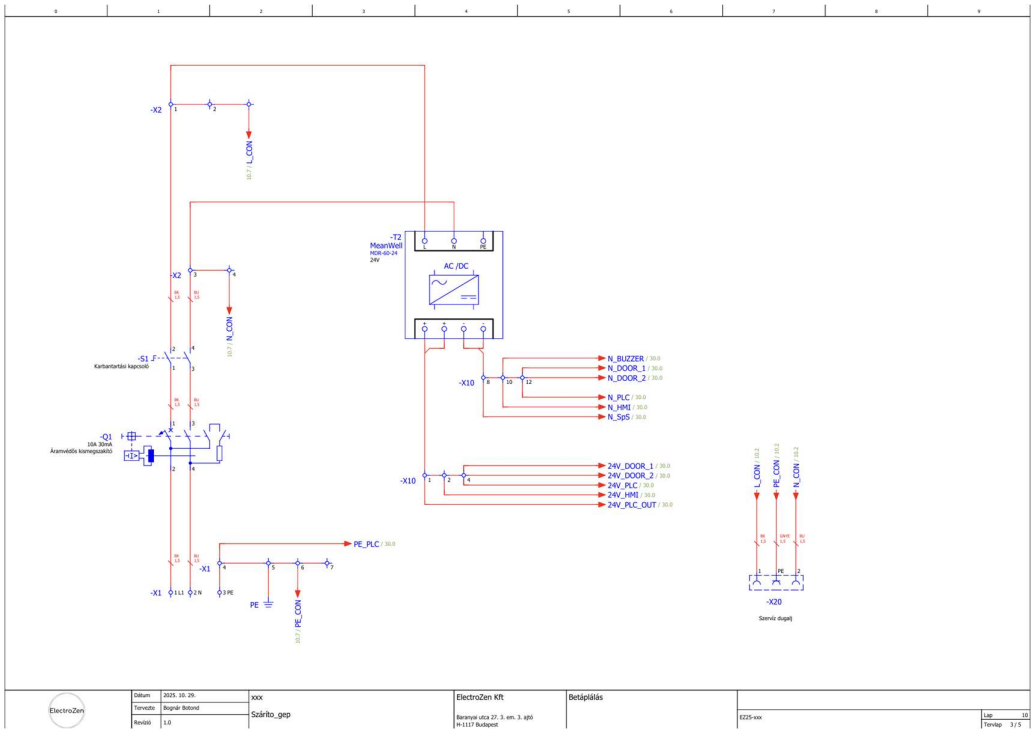
Tartalomjegyzék

X oszlop: Egy automatikusan generált tervlap kézzel került utószerkesztésre F06_001

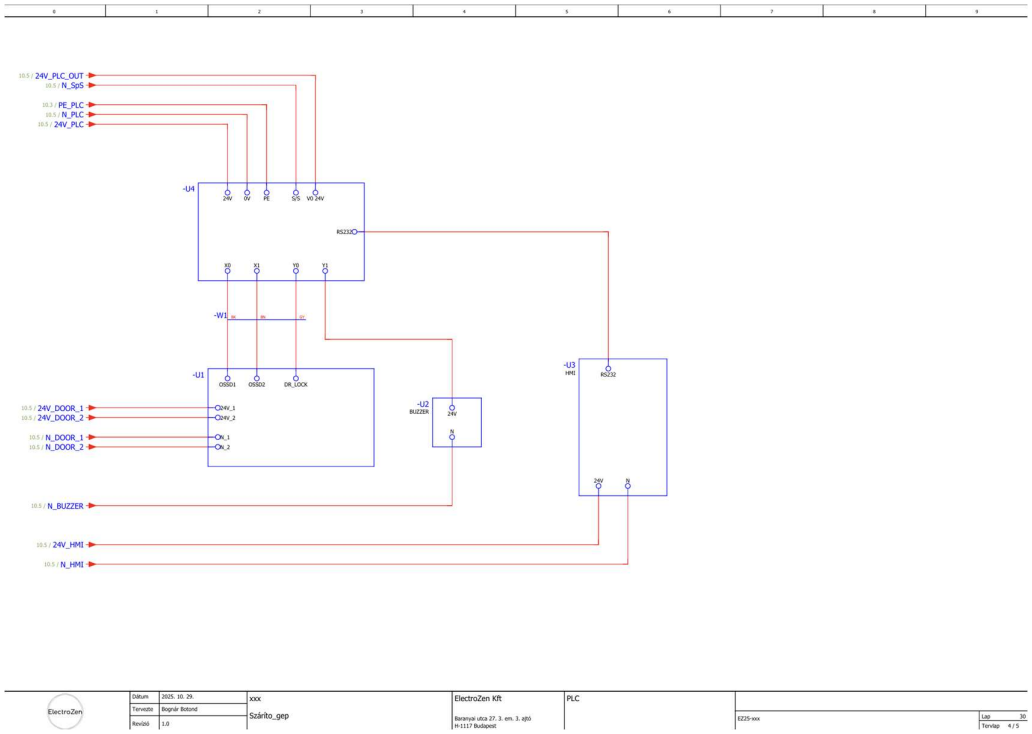
Tervlap	Tervlap leírás	Tervlap kiegészítő mező	Dátum	Szerkesztő	X
/1	Címlap / fedlap		2025. 03. 14.	Standard	
/2	Tartalomjegyzék		2025. 10. 29.	HERAN	
/10	Betáplálás		2025. 10. 29.	HERAN	
/30	PLC		2025. 10. 29.	HERAN	
/100	Alkatrész jegyzék		2025. 10. 29.	HERAN	

ElectroZen	Dátum	2025. 10. 29.	xxx	ElectroZen Kft Baranyai utca 27. 3. em. 3. ajtó H-1117 Budapest	Tartalomjegyzék
	Tervezte	Bognár Botond	Száríto_gep		
	Revízió	1.0			EZ25-xxx Lap 2 Tervlap 2 / 5

26.ábra: Elektromos terv 3.oldal



27.ábra: Elektromos terv 4.oldal



28.ábra: Elektromos terv 5.oldal

Alkatrészjegyzék

Thesis_01

Tervjel	Mennyiség	Megnevezés	Cikkszám	Szállító
-Q1	1	Áramvédős kismegszakító	R9D31210	Schneider
-S1	1	Karbantartási kapcsoló	T3-1-102/EA/SVB	EATON
-T2	1	Tápegység 24VDC	MDR-60-24	MeanWell
-U1	1	Ajtózár	DR1245	Keyence
-U2	1	Buzzer	BZ982	EATON
-U3	1	HMI	GT1455	Mitsubishi
-U4	1	PLC	FX3G	Mitsubishi
-X20	1	Dugalj	C60-DAO	Tracon

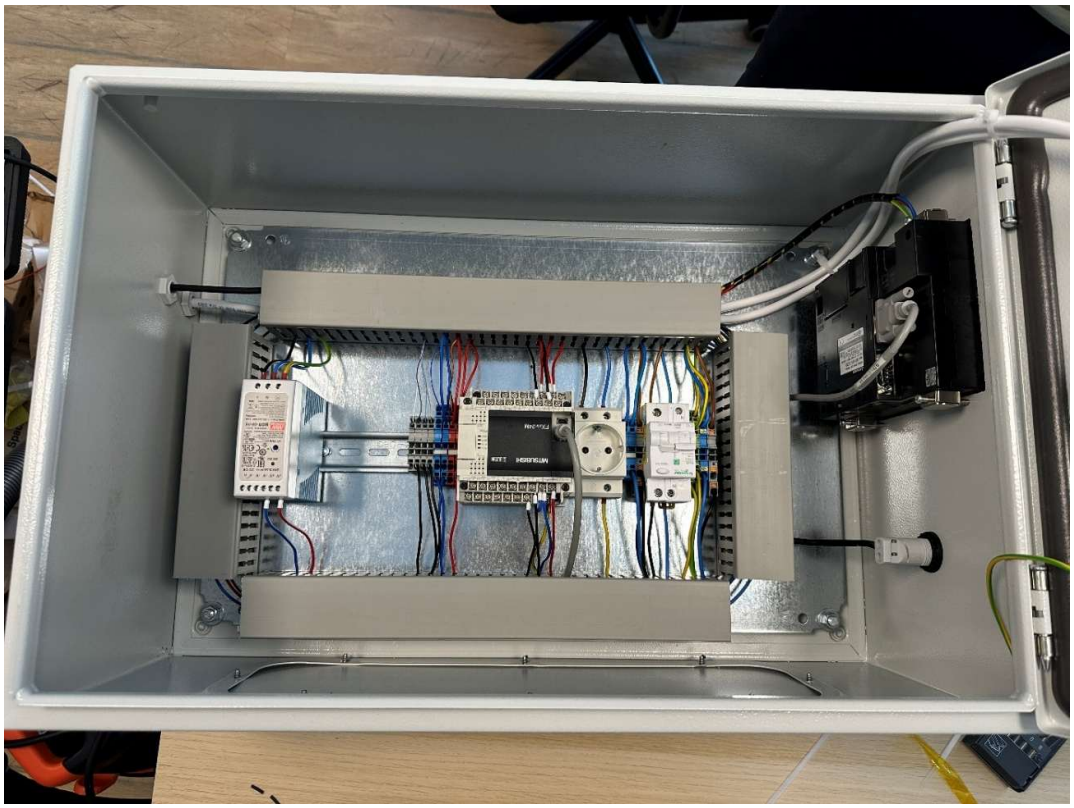
ElectroZen	Dátum	2025. 10. 29.	xxx	ElectroZen Kft	Alkatrész jegyzék	
	Tervezte	Bognár Botond	Szárító_gep	Baranyai utca 27. 3. em. 3. ajtó H-1117 Budapest		EZ25-xxx Lap 100 Tervlap 5 / 5
	Revízió	1.0				

Szent István Campus, Gödöllő
Cím: 2100 Gödöllő, Páter Károly utca 1.
Tel.: +36-28/522-000
Honlap: https://godollo.uni-mate.hu

10.3. További képek a szerelésről, berendezésről

29.ábra: Berenedezés előlnézetből

30.ábra: Berendezés belseje



31.ábra: berendezés oldalnézetből