



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

KANDÓ KÁLMÁN VILLAMOSMÉRNÖKI KAR
ELEKTRONIKAI ÉS KOMMUNIKÁCIÓS RENDSZEREK INTÉZET
MŰSZERTECHNIKAI ÉS AUTOMATIZÁLÁSI TANSZÉK



SZAKDOLGOZAT



OE-KVK
2023.

Major Miklós
T008262/F112904/K



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Óbudai Egyetem
Kandó Kálmán Villamosmérnöki Kar
Műszertechnikai és Automatizálási Intézet

SZAKDOLGOZAT FELADATLAP

Hallgató neve: Major Miklós

Szakedolgozat száma: SZD2309040923401228

Törzskönyvi száma: T008262/FI12904/K

Neptun kódja:

Szak: villamosmérnöki (magyar nyelven)

Specializáció: Műszer-automatika specializáció

A dolgozat címe: Konfekcionáló gép villamos tervezése és programozása

A dolgozat címe angolul: Electrical design and programming of a plastic film cutting machine

A feladat részletezése: A feladat egy konfekcionáló gép villamos tervének elkészítése, a megrendelővel történő egyeztetés alapján. A gép fizikai elkészülte után annak programozása és paraméterezése, az előzetes működési leírás alapján. Ezek után tesztelés a megrendelővel, illetve a megfelelő működéshez szükséges módosítások elvégzése.

Intézményi konzulens neve: Varga Árpád

A kiadott téma elévülési határideje: 2025. 12. 31.

Beadási határidő: 2023. 12. 15.

A szakdolgozat: Nem titkos.

Kiadva: Budapest, 2023. 10. 24.



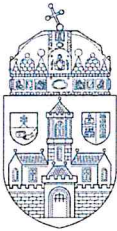
P.H

Intézetigazgató

A dolgozatot beadásra alkalmasnak találom:

Varga Árpád
belső konzulens

külső konzulens



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

KANDÓ KÁLMÁN VILLAMOSMÉRNÖKI KAR
ELEKTRONIKAI ÉS KOMMUNIKÁCIÓS RENDSZEREK INTÉZETI
MŰSZERTECHNIKAI ÉS AUTOMATIZÁLÁSI TANSZÉK



HALLGATÓI NYILATKOZAT

Alulírott hallgató kijelentem, hogy a szakdolgozat saját munkám eredménye, a felhasznált szakirodalmat és eszközöket azonosíthatóan közltem. Az elkészült szakdolgozatban található eredményeket az egyetem és a feladatot kiíró intézmény saját céljára térítés nélkül felhasználhatja, a titkosításra vonatkozó esetleges megkötések mellett.

Budapest, 2023. 12. 13.


.....
hallgató aláírása

TARTALOMJEGYZÉK

1	BEVEZETÉS	3
2	ELŐKÉSZÜLETEK, A BERENDEZÉS FELÉPÍTÉSE	4
2.1	Egyeztetés a megrendelővel	4
2.2	A berendezés felépítése	4
2.2.1	Lecsévélő egység.....	4
2.2.2	Kiegészítő egység (kivágó)	6
2.2.3	Adagoló egység	8
3	TERVEZÉS	11
3.1	A terv felépítése	11
3.2	Eszközök blokkjai a tervben.....	11
3.3	Linkek felépítése.....	12
4	PROGRAMOZÁS.....	14
4.1	Előkészületek.....	14
4.1.1	Üzem módok, üzem állapotok.....	15
4.1.2	Munkahenger vezérlés.....	16
4.1.3	Hibakezelés	20
4.1.4	Előzetes konfekcionálási ciklus	22
4.1.5	Lámpavezérlés.....	25
4.1.6	Üzemidő mérése.....	27
4.2	I-O ellenőrzés	28
4.3	Tesztelés, a gép további funkcióinak kialakítása, a program végleges felépítése	28
4.3.1	Konfekcionálási ciklus	29
4.3.2	A lecsévélő motor vezérlése	32
4.3.3	Szervohajtás	35
4.3.4	Kiegészítő funkció, markerjelfigyelés beépítése.....	40
5	ÖSSZEFOGLALÁS.....	44
5.1	Magyar.....	44

5.2 Angol.....	44
ÁBRAJEGYZÉK	46
TÁBLÁZATJEGYZÉK.....	48
KÓDJEGYZÉK.....	48
FELHASZNÁLT IRODALOM.....	49
MELLÉKLETEK.....	51

1 BEVEZETÉS

Bő egy éve dolgoztam egy ipari automatizálással, egyedi célgépek, szerelő- és gyártósorok, valamint robotcellák tervezésével és kivitelezésével foglalkozó cégnél. Az egy év ott töltött munka után jutottam el oda, hogy különböző részfeladatok megoldásán keresztül rutint szereztem a tervezési, programozási és beüzemelési folyamatokban. Ekkor kapta a cég a szakdolgozatom témáját is adó megbízást, amely egy automata konfekcionáló berendezés villamos tervezésére és programozására vonatkozott. Az eszköz különböző méretű és anyagvastagságú polietilén, polipropilén és egyéb poliészterekből készült zacskókat és zsákokat képes gyártani. Továbbá egy speciális kialakítású, fitneszgépekben használatos nejlon nadrágot, amely elősegíti a zsírégetést.

Ez volt az első önálló szakmai projektem, amelynek során a teljes tervezési és kivitelezési folyamatot önállóan vittem véghez. A munkám során minden szakmai iránymutatást megkaptam, amely a projekt megvalósításához szükséges volt. Segítségemre volt Bozi Gábor, közvetlen felettesem és mentorom. Eszközspecifikus segítséget a Siemens részéről Brechler Attila, valamint Ludvig István, és a Chemplex (a DELTA szervohajtás hazai forgalmazója) részéről pedig Keszeg Dezső nyújtott számomra.

2 ELŐKÉSZÜLETEK, A BERENDEZÉS FELÉPÍTÉSE

2.1 Egyeztetés a megrendelővel

A folyamat a megrendelővel, Kollár Györggyel történő egyeztetéssel kezdődött, aki egy személyben volt a berendezés gépész tervezője is. Ennek során került bemutatásra a gépészeti terv, és átbeszélésre a konfekcionáló működésével kapcsolatos elvárások. Továbbá egyeztettük a vezérelt elemeket és a beépítésre kerülő szenzorokat. Ennek kapcsán nehezen sikerült meggyőznünk a megrendelőt, hogy minden munkahengerre szereljenek végállásfigyelő Reed-reléket. A végfelhasználó által eddig használt konfekcionálón a munkahengereknek nincs visszaellenőrzése, kizárólag időzítéses alapon működnek. Fontos megjegyezni, hogy ez sem biztonságtechnikai szempontból, sem hatékonyságban nem megfelelő, hiszen nem követhető, hogy az adott munkahenger valóban megérkezett-e a várt pozícióra.

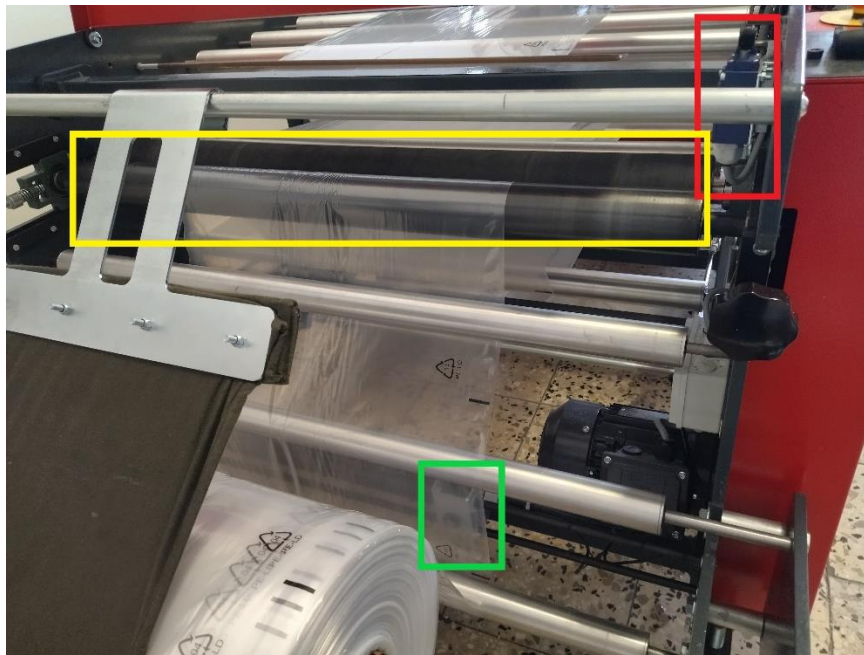
Mindezek alapján kaptunk egy előzetes működési leírást és egy előzetes I-O (input – output/ki-, bemenet) listát.

2.2 A berendezés felépítése

A berendezés 3 fő géprészre tagolható. Az alábbiakban az ezeken elhelyezkedő eszközöket fogom bemutatni.

2.2.1 Lecsévélő egység

A fólia haladási irányából nézve ez az első géprész. Ide kerül fel a fóliatekercs, amiből aztán dolgozik a berendezés. A fóliát egy szabadonfutó tengelyre erősítik fel, itt egy, a gépre akasztott, a tekercsen nyugvó, súlyozott zsák felel azért, hogy a lendület ne tekerjen le feleslegesen hosszú fóliát. Ezután a fólia meglétét jelző szenzor előtt elvezetve átfűzik két, hajtott gumihenger közt. Ezen hengereket egy frekvenciaváltóval meghajtott motor, az ezen rész után található lengőtár [2.2. ábra] állapotának és a kézi kezelőfelületen (human machine interface, továbbiakban: HMI) beállított szabályozás függvényében változó sebességgel forgatja. Így biztosítva, hogy a gépben lévő fóliát a lengőtár feszesen tartsa, de mindig legyen rajta megfelelő mennyiségű anyag a következő gyártási ciklushoz.



- Burkolat érzékelő
- Hajtott hengerpár
- Fólia érzékelő

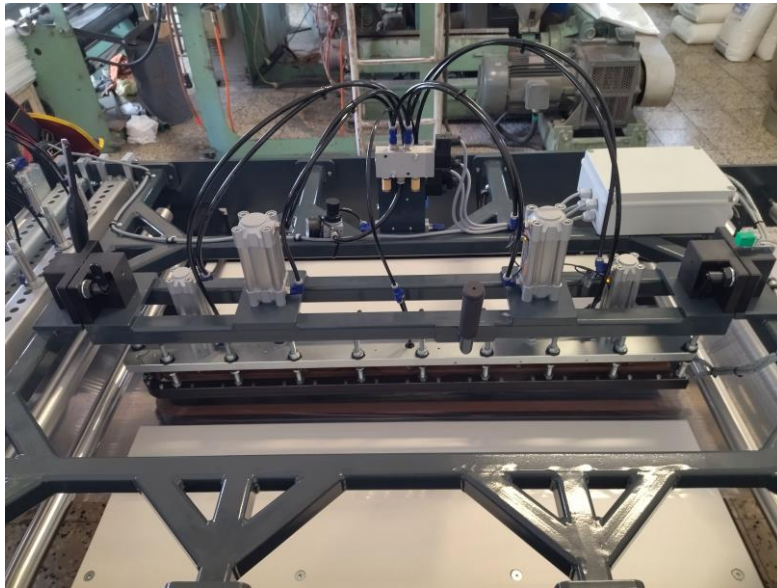
2.1. ábra – Lecsévélő [saját fotó]



- Lengőtár előfeszítő munkahenger, rajta az állapotát figyelő lengőjeladóval.

2.2. ábra - Lengőtár [saját fotó]

2.2.2 Kiegészítő egység (kivágó)



2.3. ábra - Kiegészítő egység [saját fotó]

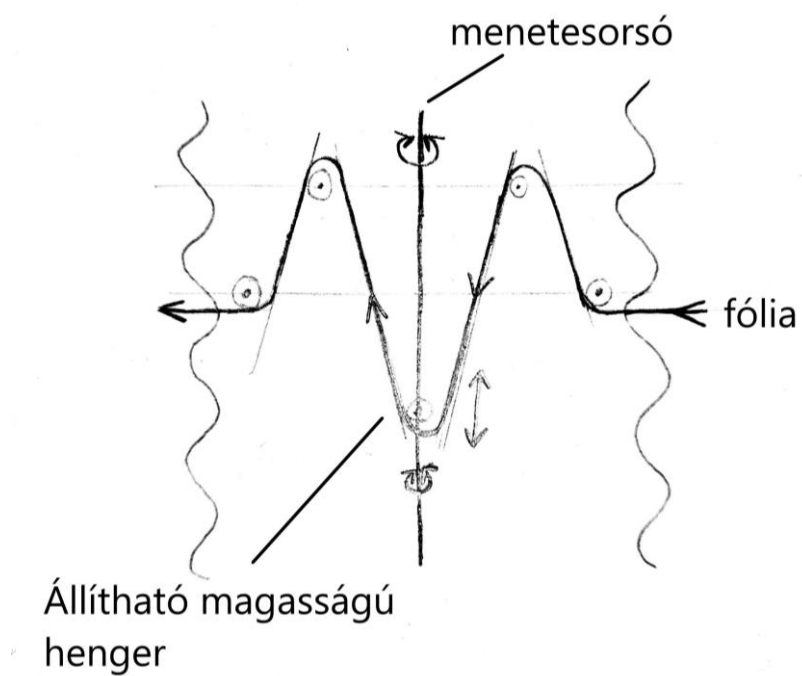
A berendezésünket elsősorban ez a része különbözteti meg más, hasonló felhasználási területű konfekcionálóktól. A képen [2.3. ábra] is látható, a fóliával párhuzamos, ahhoz képest középen elhelyezett hegesztőfej nagy szerepet kap az üzem egyik fő termékének gyártásánál. Ez a termék egy átlátszó fóliából készült nadrág, amely egy speciális, a fogást elősegítő edzőgépben használatos. A nadrág szárainak kialakítását a kivágó hegesztőfejek végzik. A felhajtható kereten pedig megtalálható a kivágó felső hegesztőfej, illetve ezen belül a leütő.

A kivágott felesleget a leütő szakítja le az anyagról, így az leesik, és egy csúszdán kicsúszik a gép mellé helyezett dobozba. Ezeket a leeső részeket, a selejtet és az esetleges egyéb hulladékokat ledarálják, és felhasználják az újabb fóliatekercsek készítésénél. A fólia a kivágó hegesztőfejeket elhagyva, a kétsoros lyukasztóba [2.4. ábra] kerül befűzésre, mikor ez szükséges. Ezeknek az egymástól, és az előttük lévő kivágótól való távolsága manuálisan állítható.

A lyukasztó és a gép harmadik nagy része közt már csak egy sor henger van. Ezek felelnek a fólia egyenletes szétterüléséért. Továbbá az egyik hengernek állítható a magassága, ezzel szabályozható a lyukasztók és az adagoló hegesztőfej közti távolság.



2.4. ábra - Lyukasztó munkahengerek [saját fotó]



2.5. ábra - Fólia felfűzése a pozícionáló hengerekre [saját fotó]

2.2.3 Adagoló egység

Az adagoló egység talán a gép legfontosabb része. Mivel itt helyezkedik el a két adagoló munkahenger – hasonlóak a gép elején találhatóéhoz, azonban itt egy szervohajtás és szervomotor gondoskodik a milliméterpontos adagolásról.

A befűzött fólia alatt és felett elvezetett kifűvók segítik az adagolást. Megfelelően beállítva úgy lebegtetik a fóliát, hogy az ne tapadjon le a már kiadagolt és levágott fóliákhoz, illetve ne is gyűrődjön fel a vége.

A kifűvók után kapnak helyet az adagoló hegesztőfejek. Ezekkel történik a fólia keresztirányú hegesztése. Bizonyos esetekben vágás is történik, a hegesztőfejre szerelt pengével. Itt a felső hegesztőfej cserélhető, így választható, hogy csak hegesztés, vagy hegesztés és vágás is történjen -e. Valamint legyen -e leszorító az adott fejen.

A gép ezen részén helyet kapott a tömbösítő, aminek a feladata a már legyártott darabokat egy kupacban tartani, hogy a következő vágás és/vagy hegesztés közben ne mozduljanak el egymásról. Ezután már csak az antisztatizáló kifűvók közt halad el a fólia, mielőtt a gép végén található lehajtható csomagolóasztalra érkezne.



2.6. ábra - Adagoló egység [saját fotó]

Mindezek mellett még része a gépnek az oldalán található kapcsolószekrény. Ezen található a HMI, amin keresztül a kezelő beállíthatja a gép működésére vonatkozó paramétereket. Itt helyezkedik még el a toronylámpa, ami a gép aktuális működési állapotáról ad visszajelzést, fény és hangjelzések segítségével. Ugyanitt található a három kezelőgomb (CIKLUS START, CIKLUS STOP, RESET), az üzemmód állító kulcsos kapcsoló és a hegesztőfejek hőfokszabályzói.

A ciklus indításért felelős CIKLUS START, és a folyamatos gyártás leállításáért felelős CIKLUS STOP gombok ki vannak párhuzamosítva egy sokméteres kábelben keresztül egy hordozható dobozra. Így, ha a kezelő személy a gép ellentétes oldalán áll vagy a felhajtható asztal végénél csomagol, onnan is könnyedén indíthatja, megállíthatja a termelést.



2.7. ábra – HMI [saját fotó]



2.8. ábra - "Kézi terminál" [saját fotó]

A HMI-vel szemközti oldalon található az adagoló kapcsoló, amivel az adagoló szervomotorját irányíthatjuk kézi üzemmódban. Itt található még néhány szelep, amelyek a hidraulikus rendszer részét képezik.



2.9. ábra - Adagoló kapcsoló [saját fotó]

3 TERVEZÉS

3.1 A terv felépítése

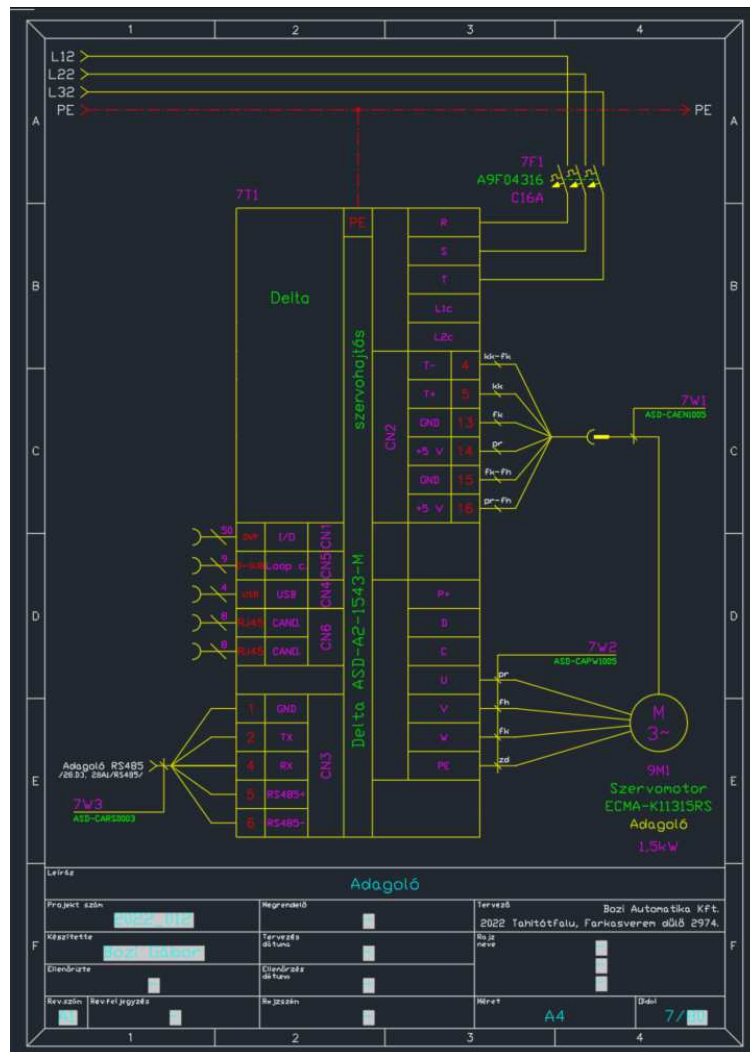
A villamos terveket AutoCAD 2017-ben készítettem, a cégnél alapvetően ezt a programot használjuk tervezésre. Van egy sablonunk, amely előre oldalakra osztja a tervet, és tartalmazza a korábban használt eszközök tervjeleit. Ebbe, az általánosan használt oldalak egyszerűen átemelhetők más, korábbi tervből. Mint például az első néhány oldal, ami a betáplálást és a szekrényt kiszolgáló eszközöket (ventilátor, szerviz dugalj, 24 V-os táp, ...) tartalmazza. Az áramút terveink általánosan a nagyfeszültségű eszközökkel kezdődnek, ezeket követik a biztonságtechnikai részek, majd a HMI és a PLC sor. Ezen eszközök után kapnak helyet először a különböző szenzorok, gombok, illetve a vezérelt eszközök, mint a munkahengerekhez tartozó mágnesszelepek, vagy a lámpák. Itt már nem köbevésett a sorrend, de igyekszünk tartani a szenzorok – érzékelők - vezérelt elemek sorrendet. A terv végén találjuk a különböző átkötéseket (a főszekrényen kívül található sorkapocssorok és a csatlakozók) és az ezekhez tartozó eszközök, az előbb említett sorrendhez ragaszkodva.

A PLC modulok összeállításánál előre kerül a CPU modul, ezt követik a digitális be-, majd kimenetek, majd az analóg egységek ugyanígy, be-, majd kimenet sorrendben. Az esetlegesen betervezésre kerülő kommunikációs modul(ok) pedig a terv ezen részének végén kapnak helyet.

3.2 Eszközök blokkjai a tervben

A tervjeleken alapvetően feltüntetésre kerülnek az adott eszköz különböző portjai, ezek közül a sorkapcsok és a nem előre gyártott kábellel használatosak, lábanként kifejtve. Ilyen esetekben az is feltüntetésre kerül, hogy az általunk betervezett kábelen belül melyik ér (kábeltípustól függően annak színe, vagy száma feltüntetésével) melyik lábra érkezik. Ebben az eszközök gyártói dokumentációjából a „wiring” fejezetei, azaz a bekötési ábrák vannak nagy segítségünkre. Ezekben a dokumentumokban, amiket a projekt Alkatrész gépkönyvek mappájában [*Alkatrész gépkönyvek*] gyűjtöttem össze általában megtalálható az eszközök minden csatlakozási pontjának részletes kifejtése. Bonyolultabb vagy sok csatlakozóval rendelkező eszközök esetében az optimális helykihasználás érdekében, gyakran csak a számunkra fontos csatlakozókat fejtjük ki részletesen. Azokat, amik az adott gépnél nem kerülnek használatra, fel sem tüntetjük a terven, ha szűkösen férnének csak el a rajzban. Feltüntetésre kerülnek még az eszközön található kapcsolók, potméterek beállításai. Egy-egy eszközt így megrajzolva külön egységként kezelhető „blokk” -ot hozunk létre. Ezeket a blokkokat illesztjük a megfelelő oldalra a készülő tervben. Jól látható, hogy az első tervváltozatban a szervohajtás blokkján [3.1. ábra] például a „CN1”, digitális ki-, bemeneti port

feltüntetésre került ugyan, de nem részleteztem az egyes sorkapcsok szerepét. Mivel úgy terveztük, hogy ez az eszköz RS-485 kommunikációs protokollt használva fogja kapni és adni a szükséges információkat a PLC felé.



3.1. ábra - Szervohajtás az első tervváltozatban [forrás: A villamos terv első revíziója]

Ezen az ábrán [3.1. ábra] is megfigyelhető, hogy minden eszköz és alkatrész egyedi tervjelet kap. Ezeknek a felépítése a következőképpen alakul: az első szám jelöli, hogy a terv hányadik oldalán találjuk az adott eszközt, ezután egy betű, vagy betűkombináció utal az alkatrész típusára; végezetül pedig még egy szám, ami azt jelzi, hogy az adott eszköz hányadik az ilyen jellegű eszközök közül az oldalon.

3.3 Linkek felépítése

Jelen tervben a vezetékek nagyrésze kivétel, nem kapott egyedi „vezeték jelölés” -t enélkül is egyértelműen beazonosíthatóak. Ez az egyértelműség a linkeknek köszönhető. Minden esetben, ahol egy vezeték, vagy kábel nem azon az oldalon végződik, mint ahol kezdődik kap egy nevet

a rajta futó jel alapján, illetve egy linket, mely a folytatását jelöli. A link tartalmazza, hogy melyik oldal melyik szegmensében találjuk a linkelt jelet, illetve, hogy az melyik eszközön hova fut be. Ezek kattintható hiperhivatkozások, így leegyszerűsítik a tervben való navigálást. PDF fájlként mentve is működnek, így az a villanyszerelő, aki mobileszközön nézi a tervet szintén könnyebben ugrik egy-egy bekötési ponthoz.

4 PROGRAMOZÁS

A Siemens PLC programozása a Siemens saját fejlesztőkörnyezetében, a TIA Portal V17 szoftverrel történt. Azért esett a választás erre a régebbi verzióra, mert a cég évek óta ezt a verziót használja, így a munkám során én is ezzel dolgoztam.

Ezen kívül az ASDA-Soft(V5.5.0.0) programot használtam, a szervohajtás felparaméterezésére. Ebből is létezik már újabb kiadás, azonban a program ezen verziójával kompatibilitási problémák adódtak. Az eszköz technikai támogatását nyújtó mérnök-üzletkötő kolléga javasoltára használtam ezt a verziót. Ezen kollégától a későbbiek folyamán is sok segítséget kaptam a hajtás felparaméterezésében.

4.1 Előkészületek

A programozás első lépéseit már a terv elkészülte után el tudtam kezdeni. Ekkor még a gép nem volt összeszerelve, emiatt fizikailag nem fértem hozzá tesztelni.

A legelső lépés a TIA -ban a „hardware config” beállítása. Ekkor, cikkszám alapján egy egyszerűen használható keresőből beimportáljuk a fizikailag is beépítésre kerülő eszközöket a programba. Ez alapján fogja tudni kezelni a programunk a különböző hálózati kártyákat, a ki- és bemeneti modulokat.

Mivel viszonylag kevés jelet használunk, egy „tag table” -ben definiáltam a ki- és bemeneteket. A terv alapján rendeltem hozzá a megfelelő I/O ponthoz a hozzá tartozó jel nevét, így a program elkészítése során egyszerűen és egyértelműen hivatkozhattam a szükséges jelre. A belső változóimat (pár kivételtől eltekintve) a jobb átláthatóság érdekében a különböző funkciókhoz rendelt adatblokkokba (Global DB) rendeztem.

A fizikai eszközök definiálása és a jelek bevitele után elkezdtem megírni a program vázát. Ebben segítségemre voltak régebbi projektek, az alapokat több projektet összevetve alakítottam ki. Egy-egy funkció, vagy funkció-blokk – mint ahogy az a későbbiekben megfigyelhető – akképpen épül föl, hogy az elején mindig a benne használt időzítők és élfigyelések kapnak helyet, ezután következnek a különböző logikai funkciók.

A „MAIN[OB1]” „organization block” -ból hívom meg a saját magam által létrehozott funkciókat.

Ebben a munkafázisban raktam össze a HMI jelentős részét is. A HMI elkészítés szintén a TIA Portalban történt, különböző be- és kimeneti mezők és a gyári recept kezelő felhasználásával.

A kinézetről és kezelésről részletesen írok a mellékelt Kezelési útmutatóban [\\Gépkönyv\2022_12_KU.PDF].

4.1.1 Üzem módok, üzem állapotok

Az első közt írtam meg a „Vezerles_FC” funkciót [4.1. kód]. Ez a kódrészlet felelős az üzem módok közti váltásért, ezen átmenetek és fenntartásának feltételeinek figyeléséért.

Ezek a sorok azért fontosak, hogy biztonságosan válthassunk a gép üzem állapotai között. Alapvetően két üzem mód van a gépen. Kézi, ami egy szerviz üzem módnak feleltethető meg, ebben az üzem módban minden munkahenger vezérelhető kézzel. Illetve az adagoló szervomotorja is indítható, kis sebességgel előre és hátra is, akár felnyitott burkolattal is. Ez az üzem mód elsősorban alapanyag cserénél használatos. Automata üzem módban pedig értelemszerűen tudjuk indítani a beállított gyártási ciklust. A felhasználó ezek közül választhat, a kulcsos üzem módválasztó kapcsolóval. Ám biztonságtechnikai szempontból fontos, hogy bármilyen hibajelenség esetén ne legyen működtethető semmi. Ezért hibaüzenet esetén egyik üzem mód sem lesz aktív. Az egyes vezérelt gépelemek vezérlésénél pedig figyelembe vesszük, hogy milyen üzem módban van a gép, így biztosítva, hogy mindig az aktuális állapotnak megfelelően működjön minden.

További biztonsági funkció, hogy az automata ciklus indításához nem elég megnyomni a start gombot, 1 másodpercig szükséges nyomva tartani azt. Ezalatt az idő alatt a kezelőnek lehetősége van újra meggyőződni róla, hogy a gép indítható, nincs semmi akadálya a biztonságos üzemnek.

```

1. //Időzítő
2. "Automata_START_kesleltetes_TON_DB".TON(IN := "START",
3.                                     PT := T#1S);
4.
5. IF      "Safety_OK"
6. AND NOT "Hibakezeles_DB".Hiba_van
7. AND NOT "Hibakezeles_DB".Figyelmeztetes_van
8. AND    "Automata_uzemmod_kapcsoló"
9. AND NOT "Kezi_uzemmod_kapcsoló"
10. THEN
11.  "Automata_uzemmod" := TRUE;
12. ELSE
13.  "Automata_uzemmod" := FALSE;
14. END_IF;
15.
16. IF      "Safety_OK"
17. AND NOT "Hibakezeles_DB".Hiba_van
18. AND NOT "Automata_uzemmod_kapcsoló"
19. AND    "Kezi_uzemmod_kapcsoló"
20. THEN
21.  "Kezi_uzemmod" := TRUE;

```

```

22. "Konfekcionalas_DB".Alaphelyzet_felvetel := TRUE;
23. ELSE
24. "Kezi_uzemmod" := FALSE;
25. END_IF;
26.
27. IF "Adagolo_hegesztesi_ido_TP_DB".Q
28. OR "Kivago_hegesztesi_ido_TP_DB".Q
29. OR "Lefuvas_TP_DB".Q
30. THEN
31. "Konfekcionalas_DB".Varakozas := TRUE;
32. ELSE
33. "Konfekcionalas_DB".Varakozas := FALSE;
34. END_IF;
35.
36. IF "Automata_uzemmod"
37. AND "Automata_START_kesleltetes_TON_DB".Q
38. AND NOT "Ciklus_STOP"
39. AND NOT "Konfekcionalas_DB".Csomag_kesz
40. AND NOT "Konfekcionalas_DB".Varakozas
41. AND NOT "Hibakezeles_DB".Hibatorles_volt
42. THEN
43. "Ciklus_START" := TRUE;
44. END_IF;
45.
46. IF ("Ciklus_STOP" AND "Ciklus_kesz")
47. OR (("Recept".Csomag_elveteli_ido = 0) AND "Konfekcionalas_DB".Csomag_kesz)
48. OR NOT "Automata_uzemmod"
49. OR "Hibakezeles_DB".Hiba_van
50. THEN
51. "Ciklus_START" := FALSE;
52. END_IF;
53.
54. IF "STOP" THEN
55. "Ciklus_STOP" := TRUE;
56. END_IF;
57.
58. IF NOT "Ciklus_START" THEN
59. "Ciklus_STOP" := FALSE;
60. END_IF;

```

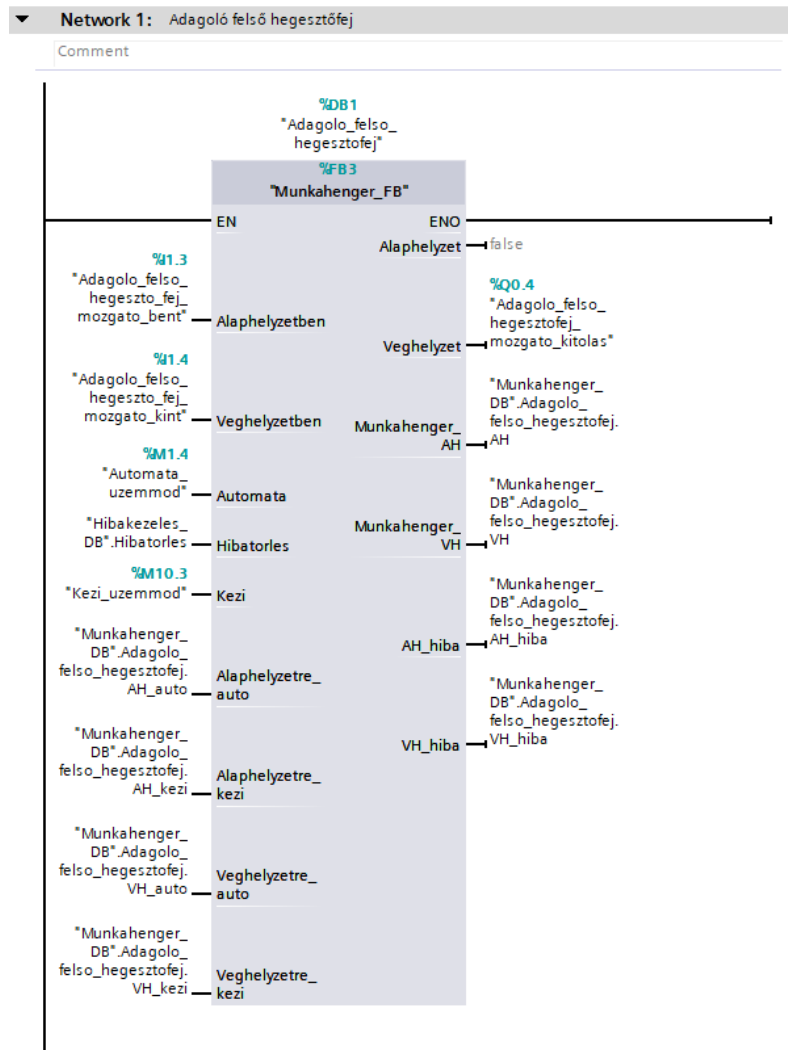
4.1. kód - „Vezérles_FC” function [saját program]

4.1.2 Munkahengervezérlés

Ebben a munkafázisban készült még el a „Munkahenger_FC” [4.1. ábra] function block is. Ebből a blokkból vezéreljük a munkahengereket, az ehhez használt „Munkahenger_FB” [4.2. kód] function block az egyetlen, amit teljes egészében módosítás nélkül vettem át egy korábbi projektből. Ezen keresztül adja ki, a beérkező parancsnak és üzemmódnak, üzemállapotnak megfelelően a hozzárendelt munkahengernek a vezérlő jelet. Az alábbi ábrán [4.1. ábra] láthatjuk, a „Munkahenger_FC” -be illesztett, az „Adagoló felső hegesztőfej” -hez tartozó munkahenger vezérléséért felelős „Munkahenger_FB” -ot az ehhez tartozó ki- és bemenetekkel összekötve. A [4.2. kód] kódrészlet betekintést enged a funkcióblokkba, így láthatjuk a működését. Ezt egészíti ki a [4.1. táblázat] táblázat, ami a "Munkahenger_FB" -hez létrehozott

DB -k belső felépítését mutatja be. Alatta pedig, a [4.1. táblázat - A "Munkahenger_FB" -hoz létrehozott DB-k belső felépítése [saját program]

] az „Adagoló felső hegesztőfej” -re vonatkozó részletet mutatja a "Munkahenger_DB" -ből. A [4.1. ábra] -n láthatjuk ennek a két táblázatnak az összerendelését, a [4.2. kód] -ban pedig a logikai összefüggéseket a blokk ki- és bemenetei között.



4.1. ábra - Részlet a "Munkahenger_FC" -ből [saját program]

```

1. #AH_hiba_idozito.TON(IN := #Automata AND NOT #AH_hiba AND (#Alaphelyzetre_kezi OR
#Alaphelyzetre_auto) AND NOT #Alaphelyzetben, PT := T#2000MS);
2. #VH_hiba_idozito.TON(IN := #Automata AND NOT #VH_hiba AND (#Veghelyzetre_kezi OR
#Veghelyzetre_auto) AND NOT #Veghelyzetben, PT := T#2000MS);
3.
4. IF NOT #Kezi THEN
5.   #Alaphelyzetre_kezi := FALSE;
6.   #Veghelyzetre_kezi := FALSE;
7. END_IF;
8.
9. IF NOT #Automata THEN
10.  #Alaphelyzetre_auto := FALSE;

```

```

11. #Veghelyzetre_auto := FALSE;
12. END_IF;
13.
14. IF (#Alaphelyzetre_auto AND #Automata) OR (#Alaphelyzetre_kezi AND #Kezi) THEN
15.   #Alaphelyzet := TRUE;
16. ELSE
17.   #Alaphelyzet := FALSE;
18. END_IF;
19.
20. IF (#Veghelyzetre_auto AND #Automata) OR (#Veghelyzetre_kezi AND #Kezi) THEN
21.   #Veghelyzet := TRUE;
22. ELSE
23.   #Veghelyzet := FALSE;
24. END_IF;
25.
26. IF #Alaphelyzetben THEN
27.   #Munkahenger_AH := TRUE;
28. ELSE
29.   #Munkahenger_AH := FALSE;
30. END_IF;
31.
32. IF #Veghelyzetben THEN
33.   #Munkahenger_VH := TRUE;
34. ELSE
35.   #Munkahenger_VH := FALSE;
36. END_IF;
37.
38. IF #AH_hiba_idozito.Q THEN
39.   #AH_hiba := TRUE;
40.   #Alaphelyzetre_auto := FALSE;
41.   #Alaphelyzetre_kezi := FALSE;
42.   #Veghelyzetre_auto := FALSE;
43.   #Veghelyzetre_kezi := FALSE;
44. END_IF;
45.
46. IF #VH_hiba_idozito.Q THEN
47.   #VH_hiba := TRUE;
48.   #Alaphelyzetre_auto := FALSE;
49.   #Alaphelyzetre_kezi := FALSE;
50.   #Veghelyzetre_auto := FALSE;
51.   #Veghelyzetre_kezi := FALSE;
52. END_IF;
53.
54. IF #Hibatorles THEN
55.   #AH_hiba := FALSE;
56.   #VH_hiba := FALSE;
57. END_IF;

```

4.2. kód - "Munkahenger_FB" [saját program]

Name	Data type	Start value	Retain
Input			
Alaphelyzetben	Bool	false	False
Veghelyzetben	Bool	false	False
Automata	Bool	false	False
Hibatorles	Bool	false	False
Kezi	Bool	false	False
Output			
Alaphelyzet	Bool	false	False
Veghelyzet	Bool	false	False
Munkahenger_AH	Bool	false	False
Munkahenger_VH	Bool	false	False
AH_hiba	Bool	false	False
VH_hiba	Bool	false	False
InOut			
Alaphelyzetre_auto	Bool	false	False
Alaphelyzetre_kezi	Bool	false	False
Veghelyzetre_auto	Bool	false	False
Veghelyzetre_kezi	Bool	false	False
Static			
AH_hiba_idozito	IEC_TIMER		False
<i>PT</i>	Time	T#0ms	False
<i>ET</i>	Time	T#0ms	False
<i>IN</i>	Bool	false	False
<i>Q</i>	Bool	false	False
VH_hiba_idozito	IEC_TIMER		False
<i>PT</i>	Time	T#0ms	False
<i>ET</i>	Time	T#0ms	False
<i>IN</i>	Bool	false	False
<i>Q</i>	Bool	false	False

4.1. táblázat - A "Munkahenger_FB" -hoz létrehozott DB-k belső felépítése [saját program]

Name	Data type	Start value	Retain
Static			
Adagolo_felso_hegesztofej	"Munkahenger_dt"		False
AH_kezi	Bool	false	False
AH_auto	Bool	false	False
VH_kezi	Bool	false	False
VH_auto	Bool	false	False
AH	Bool	false	False
VH	Bool	false	False
AH_hiba	Bool	false	False
VH_hiba	Bool	false	False

4.2. táblázat - Az „Adagoló felső hegesztőfej” -re vonatkozó részlet a "Munkahenger_DB" -ből [saját program]

4.1.3 Hibakezelés

Az alábbi két táblázatban [4.3. táblázat és 4.4. táblázat] sorolom fel a gép lehetséges hiba és figyelmeztetés állapotait. Aktív hibaüzenettel a gép sem kézi, sem automata üzemmódban nem indítható. Ettől annyiban térnek el a figyelmeztetések, hogy a kézi üzemmód funkciói azok mellett még működnek. Ebbe a kategóriába olyan állapotokat soroltam, amik veszélyessé tennék az üzemelést, de szükségesek például új anyag befűzése közben.

Hiba fellépése esetén a munkahengerek azonnal alaphelyzetre állnak, a motorok pedig lefékeződnek. A HMI-n megjelenik a hibaüzenet, amelyből kiderül, hogy mi okozta a leállást, a toronylámpa pedig pirosan kezd villogni, illetve sípoló hangjelzéssel jelez. A különböző hibák, azok megszüntetése után a kék, „Hibatörlés” gomb megnyomásával törölhetőek. Ekkor újraindítható a gyártási ciklus, az elejéről.

Természetesen a hibák listája a gép elkészültének végéig bővült, az igényeknek és tesztelés közben felmerülő gondoknak megfelelően.

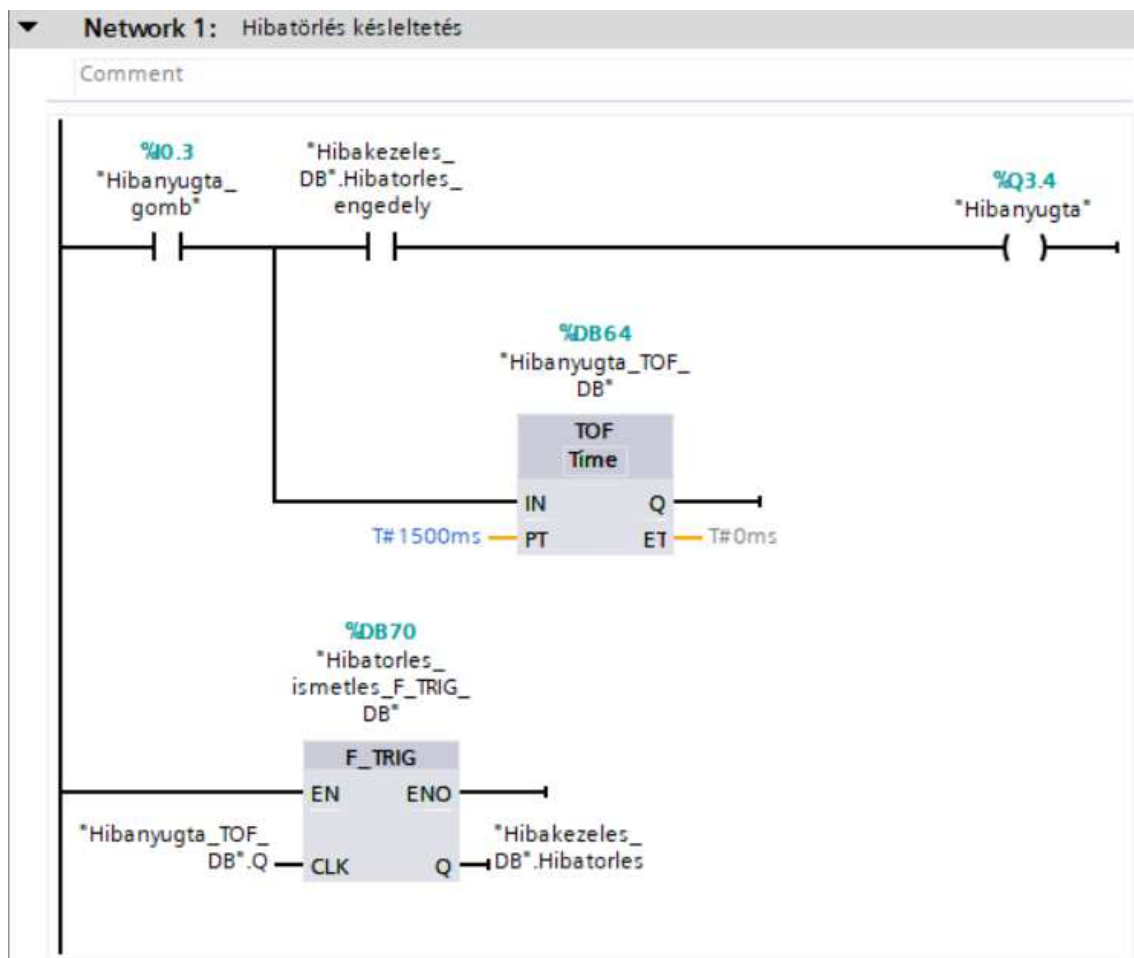
Hibaszáma	Hibaüzenet
1	Az adagoló felső hegesztőfej mozgató munkahenger alaphelyzetre állás közben elakadt!
2	Az adagoló felső hegesztőfej mozgató munkahenger véghelyzetre állás közben elakadt!
3	A burkolat nyitó munkahenger alaphelyzetre állás közben elakadt!
4	A kiütő munkahenger alaphelyzetre állás közben elakadt!
5	A kiütő munkahenger véghelyzetre állás közben elakadt!
6	A kivágó felső hegesztőfej mozgató munkahenger alaphelyzetre állás közben elakadt!
7	A kivágó felső hegesztőfej mozgató munkahenger véghelyzetre állás közben elakadt!
8	A tömbösítő munkahenger alaphelyzetre állás közben elakadt!
9	A tömbösítő munkahenger véghelyzetre állás közben elakadt!
10	A 7T1 - Adagoló szervohajtás hibára futott!
11	A 6T1 - Lecsévélő frekvenciaváltó hibára futott!
12	Vészstop benyomva!
13	Kivágó hegesztőfej kihajtva!
14	Lecsévélő burkolat nyitva!
15	A fólia begyűrődött!
16	Az alapanyag lefogyott!
17	Az adagoló burkolat nyitva van!
18	A lecsévélő sebessége 0!

4.3. táblázat – Hibalista [saját program]

Figyelmeztetés szám	Figyelmeztetés szövege
1	Lecsévélő burkolat nyitva!
2	Nincs betöltve recept! A folytatáshoz töltsön be egy receptet!

4.4. táblázat - Figyelmeztetés lista [saját táblázat]

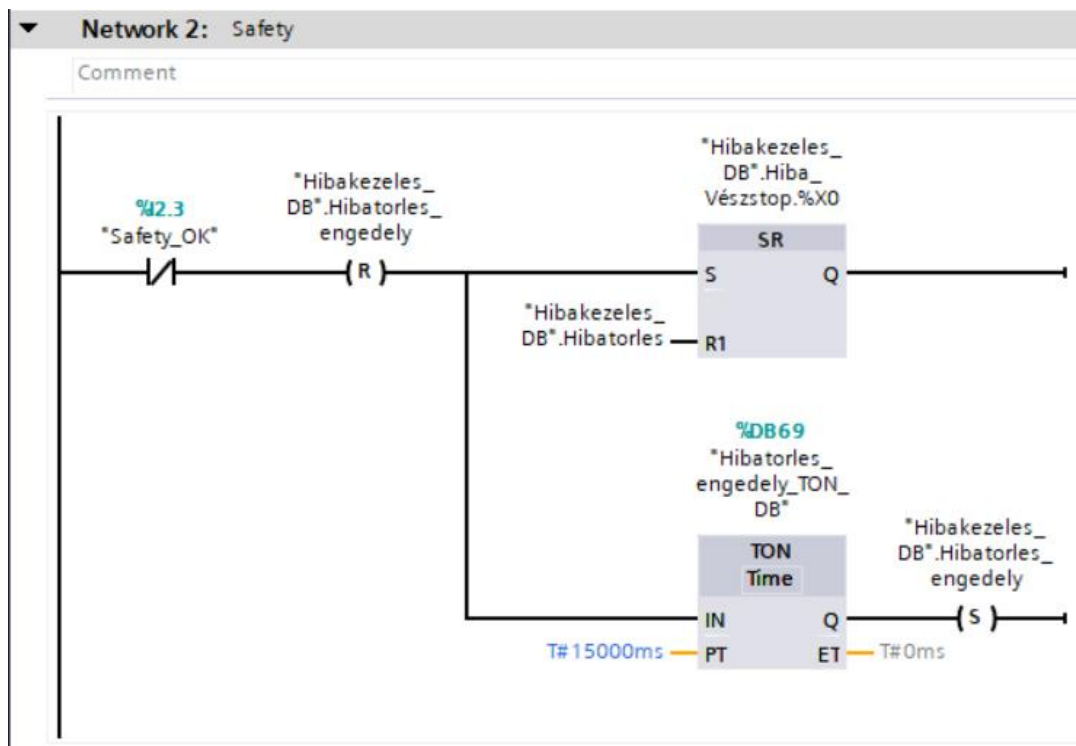
A hibák kezelése szintén különálló funkciót kapott (Hibakezeles_FC). Ennek az elején látható, hogy mi történik, ha megnyomják a hibatörles gombot [4.2. ábra].



4.2. ábra – Hibatörles [saját program]

A készletetett hibatörlésre azért van szükség, hogy miután a gomb megnyomásával felhúztuk a biztonsági relét és megérkezett a „safety_OK” jel, az ennek hiányából adódó hibát is töröljük, újabb gombnyomás nélkül. A beállított 1500 ms alatt a relé biztosan behúz. A következő kódsorban [4.3. ábra] láthatunk egy viszonylag hosszú, 15 másodperces időzítőt, aminek a lejártá előtt nem lehetséges hibát törölni. Ennek oka, hogy egy egyszerű biztonsági relés védelem van a gépen, ami vészSTOP gomb megnyomása esetén megszakítja a frekvenciaváltó

és a szervohajtás 230V-os betápját. Ilyen esetben, hogy ezeknek az eszközöknek az alkatrészei ne hibásodjanak meg, meg kell várni, hogy minden kondenzátor kisüljön bennük. Erre hivatott az alábbi védelmi idő, mely csak akkor aktiválódik, ha a „Safety_OK” jel megszűnik, azaz elveszük a fent említett eszközök áramellátását.



4.3. ábra - Hibatörles engedély [saját program]

4.1.4 Előzetes konfekcionálási ciklus

A program váza, A konfekcionálás lépéslánca a „Konfekcionalas_ciklus_FC” funkcióblokkban található. Ennek az alapját az előkészületek során írtam meg, az előzetesen megkapott folyamat leírás alapján.

„Konfekcionáló:

1. Főkapcsoló bekapcsolása, áram alá helyezzük a gépet
2. Levegő előkészítőn záró csap megnyitása, 6-8bar nyomás alá helyezzük a gépet
3. PLC megvizsgálja az alaphelyzetet, minden munkahengerek pozícióba
4. HMI látszik:
 - a. Paraméterek
 - b. Adagolás előre/hátra
 - c. Beállított darabszám/Aktuális darabszám (pl. 50/36) Csak kijelzi
 - d. Napi össz darabszám (nullázható)
 - e. Összdarabszám (nem hozzáférhető, de halványan látható bal vagy jobb felső sarokban, ez lenne az üzemóra)
 - f. Begyűródés érzékelés ki/be

- g. *Lecsévézés ki/be*
- h. *Kivágó és kiütő szelep ki/be (egyszerre használjuk, ezért együtt kapcsoljuk ki és be)*
- i. *Lyukasztó szelep ki/be*
- j. *Adagoló burkolat nyitása/zárása*
- k. *Marker jel figyelő ki/be (nem építjük ki, csak, ha esetleg majd kéri a Megrendelő egyszer akkor legyen neki helye a HMI-n.)*

5. *Állítható paraméterek:*

- a. *hegesztési idő_1 – első hegesztőfej meddig maradjon lent*
- b. *hegesztési idő_2 – hátsó „kivágó” hegesztőfej meddig maradjon lent, MEGJEGYZÉS: lehet, hogy 1 hegesztési időt állító paraméter is elég lesz az első és hátsó fejek működtetéséhez.*
- c. *hegesztés és leütés közötti idő_1 (a hegesztés indításához képest, mennyivel később legyen aktív a leütő munkahenger szelep, MEGJEGYZÉS: lehet, hogy az aktiválását a hegesztési idő lejáratához kellene kötnünk)*
- d. *hegesztés és leütés közötti idő_2 (a kivágó egység kiütő szelepét aktiválja)*
- e. *leütés ideje – NEM kell paraméterezni, csak lemegy és feljön.*
- f. *darabszám – állítási lehetőség 1- 1000-ig*
- g. *adagolási hossz – mm pontosan, 1mm – 2000mm-ig*
- h. *nyomat átlépés – Markerjel figyeléskor használjuk, a beállított hosszig nem figyel a markerjel érzékelő.*
- i. *csomag elvételei idő – 2 változat van, az első, ha az érték 0, akkor a kezelőnek kell a START gomb megnyomásával elindítani ciklust (adagolás-hegesztés-...), vagy a másik lehetőség, hogy a 1sec – 120sec között állítható az érték, ennek leteltekor automatán elindul a ciklus. Miután elérte a gép a beállított darabszámot, azután aktiválódik a csomagelvételi idő.*
- j. *hegesztés és adagolás közti idő – 0sec-10sec, miután a hegesztő munkahenger felső pozíciójába ért, akkor aktiválódik, ha az értéke nagyobb mint 0. Ez azért szokott kelleni, mert pl. a lyukasztó munkahengerek aktiválása is a hegesztési kimenetről megy, tehát addig tart míg a hegesztés, de előfordulhat, hogy a kopott lyukasztó munkahenger még nem ért a felső pozíciójába, de a hegesztő munkahenger már igen, így kezdődne az adagolás, de mivel lent van a lyukasztó munkahenger ezért ez eltépi a fóliát, ezért teszünk rá plusz időt, hogy biztos a felső pozícióban legyen.*
- k. *visszahúzási hossz – előre adagolás előtt, a beállított értékig hátrafelé adagolja a fóliát, értéke: 0 – 10mm, ha az érték nulla, akkor nincs visszahúzás. Azért szükséges, mert a fólia hegesztéskor letapadhat az alsófejre, ha így adagoljuk előre akkor begyűrődik, de ha elsőnek hátra, akkor felszakítjuk az alsófejről és adagolhatjuk előre.*

6. *Jelző torony állapotai:*

- a. *indulás – STAR gomb megnyomása után a zöld szín és a jelzőhang 2x villanása jelzi az indulást*
- b. *menet közben – zölden világít*
- c. *hiba esetén – piros lámpa villog és hangjelzés pittyeg*
- d. *készenléti állapot – sárga lámpa villog*

e. *Javaslatot elfogadok!* 😊

Konfekcionálás folyamata:

1. *feltesszük a kiinduló tekercset és befűzzük a fóliát a befűzési rajz szerint (MEGJEGYZÉS: a befűzési rajz lehetne a HMI egyik „lapján”, vagy az egyik nyomógommbal elő lehetne hívni.)*
2. *Javaslom, hogy csak kézi üzemben lehessen befűzni a fóliát, és ilyenkor csak a szükséges funkciók működjenek, mint pl.: adagolás előre/hátra lassú sebességgel, de ne lehessen elindítani az automata üzemet!*
3. *ahogy befűzzük a fóliát, először kitakarjuk a fóliával a lefogyás érzékelőt (tárgyreflexiós érzékelő), ami így aktívvá válik és engedélyezi a behúzó motort és az adagolómotort,*
4. *felnyitjuk a lecsévéző burkolatot, ami hibaiüzenetet dob a HMI-re, majd a behúzó hengerek közé befűzzük a fóliát, majd visszahajtjuk a burkolatot és töröljük a hibát,*
5. *a lengőtár pozíciójának módosításával behuzatjuk a fóliát és közben befűzzük a hengerek közé (MEGJEGYZÉS: szebb lenne, ha lenne egy nyomógomb, amivel egy fix frekvencián (frekváltó egyik bemenetét felprogramozzuk egy 10-20Hz-re) tudnánk beadagoltatni a fóliát a lengőtárba, ezt majd megbeszéljük)*
6. *elérkezünk az adagoló hengerekhez, amikor is felnyitjuk az adagoló burkolatot a HMI-ről, majd vagy a HMI-ről, vagy a másik oldalt a 3 állású kapcsolóval behuzatjuk a fóliát nagyon alacsony sebességgel, ezután lezárjuk az adagoló burkolatot és kiütjük a hibát (Megjegyzés: kézi üzem során a hibák jelzésekor a torony ne csipogjon, csak pirosan villogjon, fény és hang jelzés csak automata üzemben legyen)*
7. *kezelő belép a paraméterek menübe és beállítja azokat, adagolási hossz, hegesztési idő, darabszám..., beállítja az egyes hőfokokat és miután végzett*
8. *átállítja az üzemmód választó kulcsos kapcsolót automata üzemre és megnyomja a START gombot*
9. *a START gomb megnyomása után, ha a visszahúzás hossz>0, akkor hátrafelé elindítja a szervót a beállított hosszig, majd utána*
10. *a beállított hosszig „adagolási hossz” előre adagol,*
11. *az adagolás előtt ellenőrzi a begyűrődés érzékelőt, ha rendben van akkor engedélyezi az adagolást*
12. *csak az adagolás ideje alatt engedélyezi az antisztatizáló (ionizáló) tápegységet, valamint a kifújó szelepet,*
13. *miután leadagolta a beállított hosszt, aktiválja a hegesztési kimenetet vagy kimenteket és a lyukasztók kimenetét (5/2 szelepek),*
14. *a beállított hegesztési idő vagy idők letelnek (mindig a leghosszabb időt kel figyelni), azután aktiválódik a leütő kimenet, a reedek visszajelzése által megtörténik a fel -le mozgás és a leütő munkahengerek is alap pozíciójukba kerülnek*
15. *a biztonság kedvéért és azért, mert a lyukasztó munkahengerek pozícióit nem nézzük a program megvárja míg letelik a kezelő által beállított „hegesztés és adagolás közti idő”, így biztos, hogy a hegesztő és lyukasztó munkahengerek a felső pozíciójukba érnek. Továbbá még egy fontos dolog, ha a gépsor közepén van egy plusz hegesztő egység akkor, a hegesztés után jó, ha várunk egy kicsit mielőtt adagolnánk, mivel a meleg hegesztés könnyen megnyúlhat,*

16. ha a HMI-n engedélyezve van a kivágó egység akkor, akkor a hegesztési idő lejártakor, aktiválni kell a kiütő szelepet, valamint a lefűjő szelepet is. (MEGJEGYZÉS: előfordulhat, hogy az előbb említett kiütő és lefűjő között idő különbségnek kell lennie, ami állítható lenne, a paraméterekből),
17. miután megtörtént a leütés, kiütés és lefűjás, valamint letelt a hegesztés és adagolás közti idő, újra kezdődhet a ciklus visszahúzás – majd adagolás,
18. MEGJEGYZÉS: előzetes tervek szerint, adagolás előtt itt be kell kapcsolni egy külön szelepet, ami a levágott részt elfűjja, a bekapcsolás idejét paraméterből állíthatóvá kell megcsinálni, lehet, hogy nem lesz rá szükség!
19. a ciklusok leteltével a PLC növeli a darabszámot eggyel,
20. ahogy eléri a beállított darabszámot, a „csomag elvételi idő” aktiválódik, ilyenkor két lehetőség fut le, az egyik, hogy a kezelő a START gomb megnyomásával indítja újra a ciklust vagy a másik, hogy a beállított idő leteltével a ciklus újra indul,
21. PLC a ciklusok alatt folyamatosan figyeli a lefogyásérzékelőt, ha nem érzékeli a fűliát azonnal meg áll, nem fejezibe az aktuális folyamatot.

Elsőre ennyi, majd egyeztessünk!”

Kollár György: Folyamat leírás (2023)

Ezalapján a leírás alapján elkezdtem összerakni a gyártási ciklust. Először, a mentorommal egyeztettem. Abban állapodtunk meg, hogy az egyes munkafolyamatok egymást triggerelő időzítőkkel lesznek működtetve, amiknek az ideje HMI-ről, paraméterként állítható. Ennek alapján kezdtem el megírni az alap ciklust. Indításkor az adagolás aktiválódik, onnantól pedig sorban aktiválódnak a folyamatok, a megfelelő időtartamra és sorrendben, az egyidejűek egyszerre. Mindez eddig egyszerűen hangzik, nem is volt nagy munka a ciklus felépítése így, a leírás „Konfekcionálás folyamata” rész 9-20. pontjai alapján. A program ezen részét később teljesen át kellett alakítanom, ezért a különböző funkciók működését a [4.3.1 KONFEKCIONÁLÁSI CIKLUS] fejezetben részletezem.

4.1.5 Lámpavezérlés

Az átláthatóság kedvéért egy functionba („Lampak_FC”) gyűjtöttem a Toronylámpa és a gombok világításának logikáját. A toronylámpa hangjelzés funkciója, digitális kimenettel való spórolás miatt közösítve van a piros fényjelzés funkciójával. Alább láthatóak az idevágó részletek a kezelési útmutatóból.

„

1. A Toronylámpa funkciója és üzemállapotai:

- *Funkció: A berendezés állapotának visszajelzése.*

- *Üzemállapotok:*

1. **Pirosan villog és sípol:** A berendezés hibaállapotra futott.
2. **Sárgán villog:** A berendezésen figyelmeztetés van.
3. **Sárgán világít:** A berendezés kézi üzemmódban van.
4. **Zölden villog:** A berendezés automata üzemmódban van.
5. **Zölden világít:** A berendezés automata üzemmódban termel.
6. **Zölden és pirosan villog, sípol:** Csomag kész, új ciklus indítható.
7. **Zölden világít, pirosan villog, sípol:** Csomag kész, a csomagelvételi idő lejártá után a termelés újraindul.

2. Az adagoló kapcsoló funkciója és fényjelzései:

- *Funkció:* Az adagoló motor vezérlése.

1. **Balra fordítva:** A motor előre vezérlése.
2. **Alapállapotban:** A motor megállítása.
3. **Jobbra fordítva:** A motor hátra vezérlése.

- *Fényjelzések:*

1. **Kapcsoló nem világít:** Az adagoló motor nem vezérelhető kézzel.
2. **Kapcsoló villog:** Az adagoló motor vezérelhető kézzel.

3. Hibanyugta nyomógomb

- *Funkció:* A fennálló aktív hibák nyugtázása, valamint a biztonsági rendszer újra aktiválása.

- *Fényjelzések:*

1. **Nyomógomb nem világít:** A berendezés nincs hiba állapotban és a biztonsági rendszer aktív.
2. **Nyomógomb villog:** A berendezés hiba állapotban van, vagy a biztonsági rendszer deaktiválódott, hibatörlés szükséges.
3. **Nyomógomb folyamatosan világít:** A nyomógomb megnyomott állapotban van.

4. START nyomógomb

- *Funkció:* A berendezés automata ciklusának indítása.

- *Fényjelzések:*

1. **Nyomógomb nem világít:** A berendezés nincs automata üzemmódban. Az automata ciklus nem indítható.
2. **Nyomógomb villog:** A berendezés automata üzemmódban van. A nyomógombot hosszan nyomva tartva az automata ciklus indítható.

3. Nyomógomb folyamatosan világít: A berendezés automata üzemmódban termel.

5. STOP nyomógomb

- *Funkció:* A berendezés automata ciklusának leállítása, az éppen gyártott termék elkészülte után.
- *Fényjelzések:*

1. Nyomógomb nem világít: A berendezés automata üzemmódban termel.

2. Nyomógomb villog: Az Automata STOP funkció bekapcsolt, az automata ciklus még nem állt le.

3. Nyomógomb folyamatosan világít: A berendezés nem termel.

„

*Major Miklós: AUTOMATA KONFEKCIONÁLÓ BERENDEZÉS - KEZELÉS ÉS
ÜZEMELTETÉS (2023)*

4.1.6 Üzemidő mérése

A megrendelő kérésére hagyományos, üzemórában nem mérem az üzemidőt, csak a legyártott darabszámot figyelem. Ennek a megvalósításához létrehoztam egy „Ciklus_kesz” nevű bool típusú változót. Ennek az értékét a gyártási ciklus végén, mikor ez aktuális darab elkészült, egyre állítom., a ciklus elején, a következő darab gyártásának megkezdése előtt pedig nullázom. Az „Uzemido_FC” -ben ennek a változónak az állapotváltozását figyelem, felfutó él esetén eggyel növelem az aktuális, a napi és az összesített darabszámot.

Az „Ossz_darabszam” változó hivatott mérni, hogy hány termék készült a gépen összesen, az átadás óta. Ez a változó unsigned double int típusú, így tudja a lehető legnagyobb számot tárolni. Ez az egyetlen retail tulajdonságú változó, így ez akkor is megtartja az értékét, ha a gépet áramtalanítják. Ezt a változót nem lehet nullázni a HMI-n keresztül.

A „Napi_darabszam” változó a gép bekapcsolása, vagy a számláló utolsó nullázása óta legyártott termékek számát mutatja.

Megrendelői kérés volt, hogy lehessen beállítani csomagméretet és csomagelvételi időt. Az előre definiált csomagméret(darabszám) elkészülte után a szintén termékenként beállítható csomagelvételi idő időtartamára szüneteltetni kell a gyártást, majd mikor ez az idő lejárt, automatikusan el kell kezdeni a következő csomag készítését. Ha a csomagelvételi idő értéke

0, akkor pedig az automatikus termelést a start gomb megnyomásával lehet újraindítani. Mindkét értéket a recept paraméterek közt tudjuk beállítani.

Ez a funkció is az „Uzemido_FC” – ben kapott helyet. Az „Akt_darabszam” unsigned int típusú változóval számolom, hogy az előre beállított csomagból hány darab készült el. HMI -ről bármikor visszaállítható, új csomag kezdésekor nullázódik. Mikor az aktuális darabszám eléri a receptben beállítottat, aktiválódik a csomagelvételiidő-számláló, lejártakor pedig az aktuális darabszám értékét 0 -ra változtatom, Ezzel újraindítva a gyártást a következő csomag elkészültéig.

4.2 I-O ellenőrzés

Az előző fejezetben ismertetett programrészleteket még azelőtt írtam meg, hogy a gép fizikailag összeépítésre került volna. Így tesztelési lehetőség sem állt rendelkezésemre.

Mikor a gép elkészült és a villanyszerelő is befejezte rajta a munkát, átvehettem a terepet, és elkezdhettem tesztelni az eddig megírt programrészleteimet. Ezelőtt viszont volt még egy nagyon fontos teendőm: le kellett ellenőriznem, hogy minden a terv szerint lett -e bekötve. Ehhez a TIA Portal „watch table” funkcióját használtam. Előzetesen minden ki- és bemenetet felvettem ide, így mikor rácsatlakoztam a PLC -re és online-ra állítottam a kapcsolatot, itt láthattam valós időben a ki- és bemenetek állapotát.

Először a bemeneteket teszteltem. Ebben segítségemre volt a villanyszerelő, aki a különböző szenzorok aktiválásával jelzésre készítette azokat, így én a Tia Portalban láthattam, hogy melyik bemenetre érkezik be a jele. Mikor mindent rendben találtunk, a kimenetekkel folytattam. Ekkor ellenőriztük, hogyha kiadok egy jelet, akkor annak megfelelő vezérelt elemet aktivál-e (Munkahengerek, lámpák, stb...). Így megbizonyosodtam róla, hogy a programban létrehozott változóim megfelelnek a tényleges funkciójuknak.

Ezeket túl adódott még tervjavítási/módosítási munka is, ugyanis ekkor kaptam meg a villanyszerelőtől a tervet, amiben a hibáimat javította, illetve ahol eltért a tervtől, azt jelezte. A mellékletek közt megtalálhatóak a változtatott oldalakról készült fotók, amik alapján módosítottam a CAD fájlt.

4.3 Tesztelés, a gép további funkcióinak kialakítása, a program végleges felépítése

A bekötések ellenőrzése után megkezdtem a már kész programrészletek tesztelését, majd a még hiányzó funkciók megírását, valamint a frekvenciaváltó és a szervohajtás felparaméterezését és a kommunikáció kiépítését. Ezeknek a kialakításáról, illetve a menet közben előforduló

elakadásokról és azok megoldásáról számolok be a következő fejezetekben. A gép kezeléséről a csatolt Kezelési útmutatóban [\\Gépkönyv\2022_012_KU.pdf] olvashatnak bővebben.

4.3.1 Konfekcionálási ciklus

Mikor megmutattam a megrendelőnek az előzetes működési leírás alapján megírt konfekcionálási ciklust, hamar kiderült, hogy abban nem minden volt egyértelmű, vagy kellően részletes. Sok munkafázisnak az időzítése nem az előtte lévőtől kell függjön, mert azzal egyidőben, vagy annak lejárta előtt is indíthatónak kell lennie. Ezért úgy döntöttem, hogy a korábbi egymást triggerelő időzítős megoldás helyett lépésláncot hozok létre a folyamat nyomonkövetésére. Csak így lesz megfelelően átlátható a program és követhető, hogy az éppen futó részfeladatok közül melyik járt már le, és melyik nem. Egyes lépések végén így is hosszú, sok feltételt vizsgáló elágazások vannak. Például a lépéslánc 60. lépésének [4.3. kód] végén. Ebbe az állapotba akkor kerül a folyamat, ha elindult a kivágó hegesztés és a lyukasztás, de nem indult még el az adagoló hegesztés. Innen kezdem indítani a kiütést és a lefűvást.

```

412.      60: // Adagoló hegesztés indítás, Kivago hegesztés, Kiütés indítás, Lefűvás indítás, Lyukasztás
413.      "Antiszaizalo_tapegyseg_engedelyezes" := FALSE;
414.      "Konfekcionalas_DB".Kifuvas := FALSE;
415.      "Konfekcionalas_DB".Tombosites := TRUE;
416.      "Konfekcionalas_DB".Adagolo_hegesztes_inditas := TRUE;
417.      "Konfekcionalas_DB".Leutes_inditas := FALSE;
418.      "Konfekcionalas_DB".Kivago_hegesztes_inditas := FALSE;
419.      "Konfekcionalas_DB".Kiutes_inditas := TRUE;
420.      "Konfekcionalas_DB".Lefuvas_inditas := TRUE;
421.      "Konfekcionalas_DB".Lyukasztas_inditas := FALSE;
422.
423.      IF NOT "Munkahenger_DB".Adagolo_felso_hegesztofej.VH AND NOT
         "Munkahenger_DB".Kiuto.VH AND "Munkahenger_DB".Lefuvo.VH THEN
424.      "Konfekcionalas_DB".Lepesszam := 70;
425.      ELSIF NOT "Munkahenger_DB".Adagolo_felso_hegesztofej.VH AND
         "Munkahenger_DB".Kiuto.VH AND NOT "Munkahenger_DB".Lefuvo.VH THEN
426.      "Konfekcionalas_DB".Lepesszam := 80;
427.      ELSIF NOT "Munkahenger_DB".Adagolo_felso_hegesztofej.VH AND
         "Munkahenger_DB".Kiuto.VH AND "Munkahenger_DB".Lefuvo.VH THEN
428.      "Konfekcionalas_DB".Lepesszam := 90;
429.      ELSIF "Munkahenger_DB".Adagolo_felso_hegesztofej.VH AND NOT
         "Munkahenger_DB".Kiuto.VH AND NOT "Munkahenger_DB".Lefuvo.VH THEN
430.      "Konfekcionalas_DB".Lepesszam := 120;
431.      ELSIF "Munkahenger_DB".Adagolo_felso_hegesztofej.VH AND NOT
         "Munkahenger_DB".Kiuto.VH AND "Munkahenger_DB".Lefuvo.VH THEN
432.      "Konfekcionalas_DB".Lepesszam := 130;
433.      ELSIF "Munkahenger_DB".Adagolo_felso_hegesztofej.VH AND "Munkahenger_DB".Kiuto.VH
         AND NOT "Munkahenger_DB".Lefuvo.VH THEN
434.      "Konfekcionalas_DB".Lepesszam := 140;
435.      ELSIF "Munkahenger_DB".Adagolo_felso_hegesztofej.VH AND "Munkahenger_DB".Kiuto.VH
         AND "Munkahenger_DB".Lefuvo.VH THEN
436.      "Konfekcionalas_DB".Lepesszam := 150;
437.      END_IF;

```

4.3. kód – Egy lépés a „Konfekcionalas_ciklus_FC”-ből [saját program]

A lépésláncot „CASE szerkezet” -be rendezve valósítottam meg. Ezen a kódrészleten pedig jól látszik a lépések általános felépítése. A lépés száma mellé írtam, hogy mi történik az adott lépésben. A benne található kódban pedig fixen helyet kapnak a munkahengerek és szelepek által végzett munkafolyamatokhoz tartozó időzítők indítása/tiltása. Ezzel biztosítom, hogy minden folyamat az aktuális lépésnek megfelelően lesz –e végrehajtva. Ezen kívül, amelyik lépésben szükséges, az elején található egyéb eszközök (pl begyűrődés figyelő, szervó, ...) vezérlése. A kódrészlet legvégén mindig a következő állapotba lépés feltételeit találjuk „IF” feltételvizsgálattal, szükség esetén „ELSIF” elágazásokkal. A jobb követhetőség érdekében minden esetben a kód vége felé haladunk, soha nem léptetem kisebb sorszámú állapotba a gépet, mint az aktuális, így is csökkentve a hibalehetőségeket, és növelve az átláthatóságot. Értelemszerűen ezalól az utolsó lépés kivételt képez, ahonnan, miután a gyártási ciklus befejeződött és minden biztonsági idő lejárt, a 0., indításért felelős lépésbe ugrik a ciklus. Innen, az alaphelyzet és a szervóba írt paraméterek ellenőrzése után indul a következő darab legyártása.

A „Konfekcionalas_ciklus_FC” funkcióban, még a program alapját képező lépéslánc előtt megtalálhatóak a különböző folyamatok elvégzéséhez szükséges időzítők. Az adagoló hegesztéshez tartozón fogom most bemutatni egy-egy ilyen folyamat működését. Mint a lenti kódrészletekből [4.4. kód] látható, a lépésláncból csak a „Adagolo_hegesztes_inditas” változó TRUE állapotba állításával aktiválom a folyamatot, ami alapján aztán az adagoló munkahengert mozgatják az időzítők.

Az időzítők aktiválódási sorrendben:

- „Adagolo_hegesztes_kesleltetes_TON_DB”.TON
Az adagolás és a hegesztés közti várakozási idő. Ezalatt a recept paraméterként beállítható idő alatt megvárjuk, hogy a lendület miatt túlcusúzó fólia a lengőtár feszítésének hatására a helyére rendeződjön.
- "Adagolo_hegesztesi_ido_TP_DB".TP
Ezt az impulzust az előző TON aktiválódása indítja. Hossza szintén az adott termékre vonatkozó paraméter. A munkahenger végállásra vezérlése ehhez az impulzushoz van kötve, amint lejár, a munkahenger visszaáll alaphelyzetre.
- "Adagolo_hegesztesi_ido_vege_F_TRIG_DB"
Az előző impulzus lefutó élet jelzi. Aktív státusza bebillenti az

„Adagolo_hegesztes_vege” változót, ami pedig a „Hegesztes_vege” változó állapotára lesz hatással.

- "Adagolo_hegesztes_utani_ido_TP_DB".TP

Mivel a friss, még meleg hegesztés könnyen megnyúlik, ezért ide beiktattam még egy biztonsági időt. Szintén receptben állítható a hossza, lejártá előtt nem indítható adagolás.

```
"Adagolo_egesztes_kesleltetes_TON_DB".TON(IN:="Konfekcionalas_DB".Adagolo_hegesztes_inditas,
PT:="Recept".Hegesztes_kesleltetes_ido *100);
```

```
"Adagolo_hegesztesi_ido_TP_DB".TP(IN := "Adagolo_egesztes_kesleltetes_TON_DB".Q,
PT := "Recept".Adagolo_hegesztesi_ido * 100);
"Adagolo_hegesztes_utani_ido_TP_DB".TP(IN := "Adagolo_hegesztesi_ido_vege_F_TRIG_DB".Q,
PT := "Recept".Adagolo_hegesztes_utani_ido * 100);
```

```
"Adagolo_hegesztesi_ido_vege_F_TRIG_DB"(CLK:="Adagolo_hegesztesi_ido_TP_DB".Q);
```

```
"Hegesztes_vege_F_TRIG_DB"(CLK := "Konfekcionalas_DB".Hegesztes_vege);
```

```
IF "Adagolo_hegesztesi_ido_vege_F_TRIG_DB".Q THEN
  "Konfekcionalas_DB".Adagolo_hegesztes_vege := TRUE;
END_IF;
```

```
IF "Konfekcionalas_DB".Adagolo_hegesztes_vege THEN
  IF NOT "Recept".Kivago_aktiv THEN
    "Konfekcionalas_DB".Hegesztes_vege := TRUE;
  ELSIF "Recept".Kivago_aktiv AND "Konfekcionalas_DB".Kivago_hegesztes_vege THEN
    "Konfekcionalas_DB".Hegesztes_vege := TRUE;
  END_IF;
ELSE
  "Konfekcionalas_DB".Hegesztes_vege := FALSE;
END_IF;
```

```
//Munkahenger vezérlések
"Munkahenger_DB".Adagolo_felso_hegesztofej.AH_auto := NOT "Adagolo_hegesztesi_ido_TP_DB".Q;
"Munkahenger_DB".Adagolo_felso_hegesztofej.VH_auto := "Adagolo_hegesztesi_ido_TP_DB".Q;
```

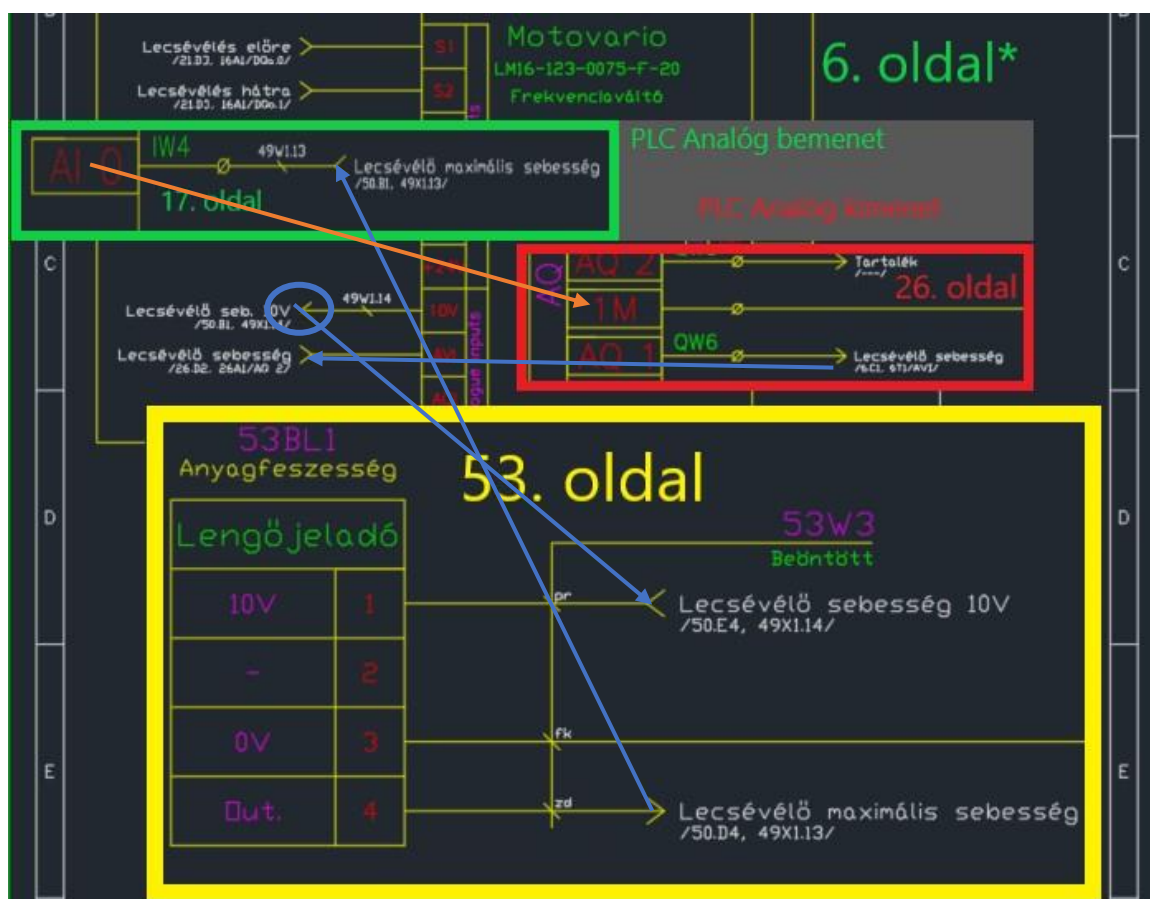
4.4. kód - Munkahengervezérlés a "Konfekcionalas_ciklus_FC" -ből [saját program]

Az időzítőkön kívül található még itt egy biztonsági funkció, ami az automata üzemmódból való kilépéskor minden munkahengert visszavezérel alaphelyzetre, a motorokat megállítja, a kommunikáción keresztül kiadott pozíciókat és parancsokat visszatörli. Illetve itt vannak az automata üzemmódba álláskor való alaphelyzet felvételért felelős sorok is. Ezekben a szervot állítom „home” pozícióra, illetve a munkahengereket alaphelyzetre.

4.3.2 A lecsévéző motor vezérlése

A legelső elképzelés szerint mindkét motort a munkahengerekéhez hasonlóan kialakított „Motor_FB” function block -on keresztül szerettem volna megvalósítani, azonban hamar rá kellett jönnöm, hogy a szervomotorhoz ez a módszer nem lesz megfelelő. Erről a [4.3.3 SZERVOHAJTÁS] cím alatt írok részletesebben.

Az alábbi képen [4.4. ábra] látható a lengőtárra szerelt analóg jeladó bekötése. Mivel a frekvenciaváltó analóg bemenete 0-10V-os szabályzó jelet vár, ezért a saját 10V-os kimenetét használtam a jeladó bemenetére. Maga a jeladó REED-relék segítségével, a lengőtárat feszítő munkahenger aktuális állásával arányosan csökkenti a feszültséget a kimenetén. Így fokozatosan csökkentve a lecsévéző motor sebességét a lengőtár telített végállása felé közeledve.



4.4. ábra - A 6T1 -Lecsévéző frekvenciaváltó sebességszabályozása¹ [saját terv]

A vevő által elvárt működés, olyan formában valósul meg, hogy ezt a sebességet a kezelő még kézzel tudja állítani. Így elérve, hogy az aktuális gyártási program ciklusidejéhez és

¹Az ábrán feltüntetett oldalszámok a 2022_012_A3 számú áramúterv megfelelő oldalára vonatkoznak.

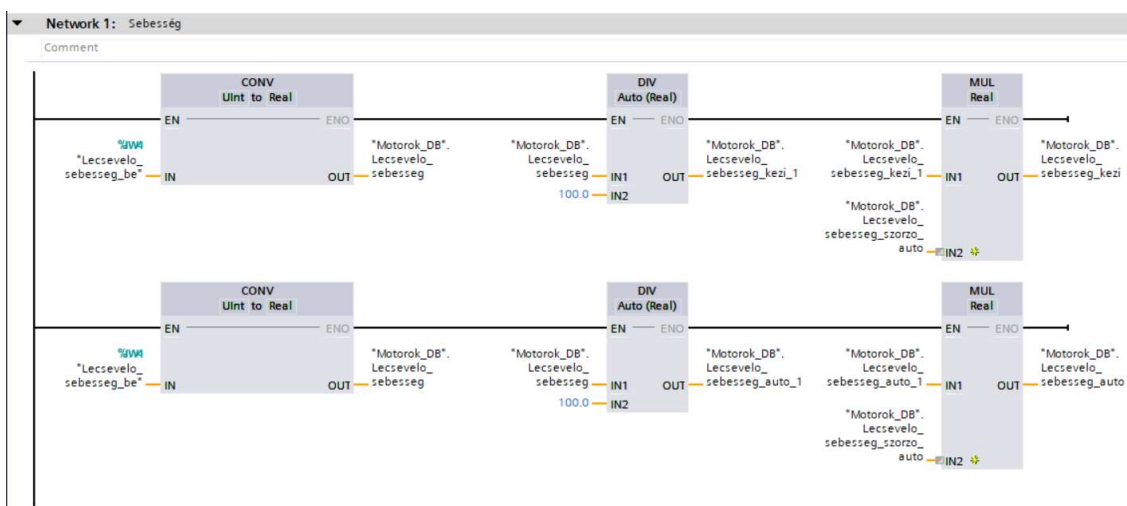
termékhosszához úgy állítsa be a lengőtár feltöltését, hogy azon mindig elegendő nyersanyag legyen előkészítve, de a motor folyamatos adagolása mellett soha ne érje el a telített végállást. A folyamatos adagolás biztosítja, hogy a gép végére akasztott nyersanyag göngyöleget nem rántja meg a motor, és nem is áll le hirtelen, amikor pedig a tekercs lendülete miatt további fólia tekeredhetne le. Azzal, hogy nem éri el a telített végállást a lengőtár, azt biztosítjuk, hogy a gépben lévő fólia mindig feszes marad. Ezt a funkciót úgy valósítottam meg, hogy a jeladóból érkező jelet a PLC analóg bemenetére kötöttem, majd a képen [4.5. ábra] látható módon a HMI -n százalékosan megadható értékkel átszámolva, adom tovább a megfelelő funkcióblokknak [4.6. ábra], ahonnan, amikor a feltételek adottak, az érték egy analóg kimeneten továbbítódik a frekvenciaváltó felé.

A frekvenciaváltó felparaméterezéséhez nem használtam külön programot, vagy külső eszközt, a szükséges beállítások elvégzéséhez elegendő volt az eszköz saját kijelzője és gombjai. Az elvégzett paraméter beállításokat a 2022_012_GI.xlsx file-ban dokumentáltam [4.5. táblázat].

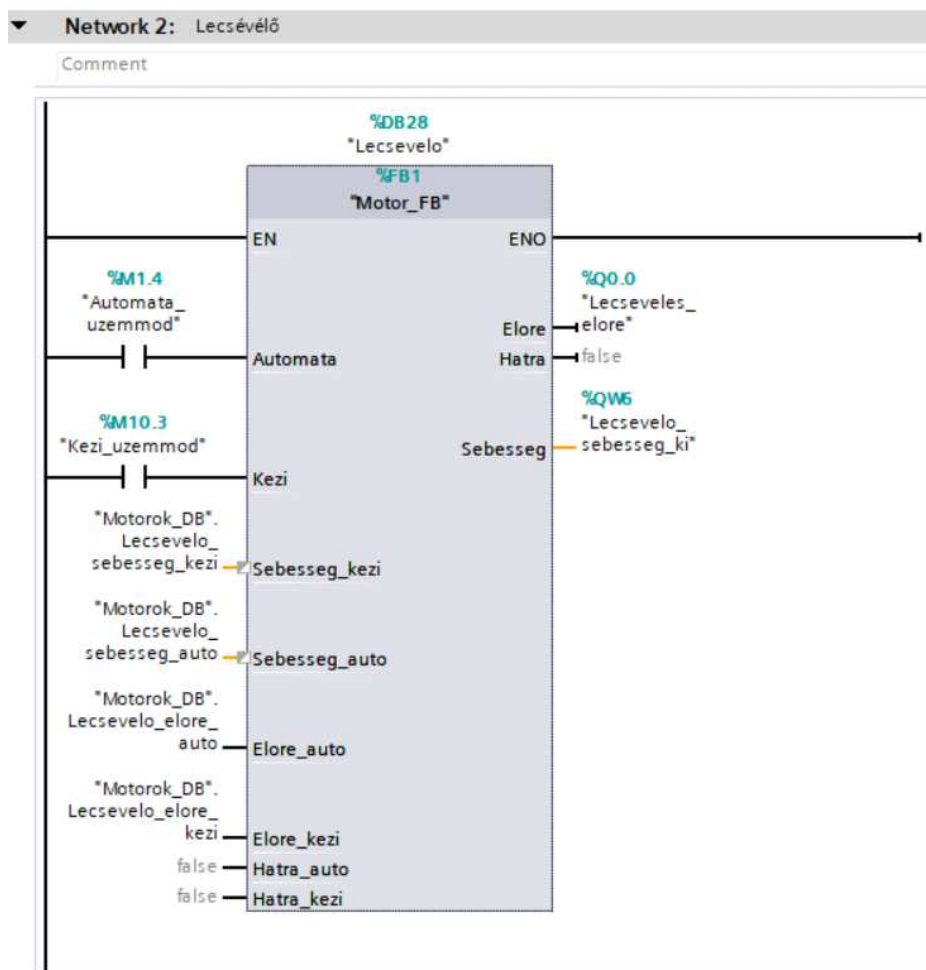
A gép kezelőjének különös körültekintéssel kell beállítani a sebesség szabályozás mértékét, ugyanis, ha túl lassan telik újra a lengőtár, előfordulhat, hogy elfogy róla a nyersanyag, és az adagolásnál meghúzzuk a fóliát. Ide valószínűleg érdemes lett volna valamilyen védelmet beépíteni, de ez az igény senki részéről nem merült fel.

6T1		
Paraméter	Érték	Funkció
00-01	1	Motor direction
00-02	1	Main run source
00-04	0	Operation mode
00-05	2	Frequency source
00-12	50	Frequency upper limit
00-14	5.0	Aceleration time
00-15	1.0	Deceleration time
01-02	50	Max frequency
02-00	2,5	Motor rated current
02-03	1420	Motor rated speed
02-04	230	Motor rated voltage
02-05	550	Motor rated power
02-06	50	Motor rated frequency
03-00	1	S2-Reverse
03-01	0	S1-Forward
03-02	17	S3-Reset
03-11	1	Relay-Fault
04-00	0	AVI-0-10V
13-08	1250	Factory settings
04-03	0	BIAS
04-04	1	Bias selection

4.5. táblázat - 6T1-Lecsévlő frekvenciaváltó felparaméterezése [saját táblázat]



4.5. ábra - Lecsévlő sebesség visszaosztása [saját program]



4.6. ábra - A Lecsévlő motor funkció blokkja [saját program]

4.3.3 Szervohajtás

A legtöbb nehézséget ez az eszköz okozta, mivel egy számunkra eddig ismeretlen, kínai márkát választott a megrendelőnk. A Delta ASDA-A2 családjából származó modellel dolgoztunk.

A szervoval való munkafolyamat ott kezdődött, hogy nem küldtek hozzá megfelelő kábelt, amivel a „CN3” -as csatlakozójával kompatibilis lenne. Mikor felvettük a kapcsolatot a magyarországi forgalmazóval, küldtek egy gyári kábelt, amit csatlakoztatni lehetett volna a szervohoz, de a másik végén egy D-SUB csatlakozót találtunk, a mi PLC RS-485 kommunikációért felelős modulunk pedig sorkapocssorral van ellátva. Ez volt az egyetlen elérhető kábel a készülékhez. Miután sehol sem találtunk információkat a kábel belső érkiosztásáról, kaptunk egy csatlakozóátalakítót, ami lehetővé teszi, hogy RJ-45 csatlakozóval szerelt kábellel csatlakozzunk a servo „CN3” -as, RS-485 kommunikációért felelős portjához. Ez a folyamat sok időbe került, de ezalatt tudtam foglalkozni a program többi részének véglegesítésével.

A legelső feladat szervó beállításával kapcsolatban, hogy beállítsuk a mozgásra vonatkozó paramétereket, a mozgatott dolgok függvényében. Ebben a Delta saját szoftverének, a Delta softnak a beépített „auto gain tuning” nevet viselő része volt segítségemre. Ebből a toolból, terhelés alatt JOGolva a motort, kiszámolja nekünk a szükséges adatokat. Ezeknek az értékeit pedig egy gombnyomással bevihetjük az eszköz memóriájába.

Az előbb ismertetett folyamat után már elkezdhetjük beállítani a szervó többi paraméterét.

7T1		
Paraméter	Érték	Funkció
P0-01	olvas	Alarm display
P0-46	olvas	Digital status display
P1-01	0x0101	forgásirány, control mód
P1-44	2560000	Numerator 1
P1-45	257	Denominator
P2-08	10	FACTORY RESET
P2-10	0x0101	DI1-Servo on
P2-11	0x0102	DI2-Alarm reset
P2-14-17	0x0100	DI5-8-Disabled
P3-01	0x0005	Transmission speed
P3-02	0x0006	data bits, parity, stop bit
P3-05	0x0001	mechanism
P3-06	írni kell	Digital Input Communication Function
P4-05	írni	JOG controll
P4-07	Írni, olvasni	Multi function of DI
P5-04	0x0008	Homing mode
P5-07	Írni, olvasni	Trigger position command
P5-20	2000	Acceleration
P5-21	50	Deceleration
P5-60	írni	SPEED
P6-00	0x00000000	Homing definition
P6-01	30000	Origin definition
P6-02	0x00001003	Position 1 definition
P6-03	Írni	Position 1 data
P6-04	0x00001002	Position 2 definition
P6-05	Írni	Position 2 data
P2-18	0x115	DO1-Target pos elérve
P2-18	0x105	DO2-Motor 0 sebességben
P2-12	0x0101	DI3-Servo ON
P2-13	0x0137	DI4-JOG Forward

4.6. táblázat - A szervohajtás általam beállított paraméterei [saját táblázat]

Először azon elgondolás szerint haladtam, hogy a szervóval való minden kommunikációt az RS-485 -ön keresztül valósítok meg. Mivel ebben még nem volt tapasztalatom, ezért a

siemenstől kértem technikai segítséget. Szerencsére készségesen válaszoltak a kérdéseimre, útmutatást adtak a PLC beállításához, és kaptam egy mintaprogramot is, ami alapján összerakhattam a saját „Kommunikacio_FC” funkciómát. A szakmai útmutatások segítségével már viszonylag könnyen fel tudtam éleszteni a kommunikációt a két eszköz között.

Úgy döntöttem, a szervo pozíció vezérelt (továbbiakban: PR) üzemmódját fogom használni, hiszen előre beállított hosszra kell adagolni. Ez úgy működik, hogy előre definiálunk pozíciókat a P6 -os paramétercsoportban. Minden pozícióhoz két paraméter tartozik, egy definíció és egy érték. Ezek beállításában a felugró Parameter Setting Helper van a segítségünkre. Ezeket a pozíciókat pedig digitális ki- és bemeneten vagy soros kommunikáción keresztül lehet triggerelni. Ennek hatására az adott pozíció beállításainak megfelelően a pozícióra áll a motor. Működése a következőképpen alakul: soros kommunikáció használata esetén a P5-07 paraméter értékét a triggerelni kívánt pozíció sorszámát. Ugyanezen a paraméteren keresztül kaphatunk visszajelzést a motor állapotáról. Az eredetileg beírt érték + 10000 jelenti, hogy a motor a megadott pozícióra mozog. Mikor eléri azt, a paraméter értéke 20000 lesz.

Parameter Name	Unit	Minimum ~ Maximum	Default	16/32 bit
P6 - 02		0x00000000 ~ 0xFFFFFFFF	0x00000000	32bit

PATH#1 Definition

Value: 0x00001003

TYPE settings [3]: Auto positioning control. Motor goes to the next path when positioning is complete.

OPT options
 INS: When this PR is executing, it will interrupt the previous
☒ 0:NO ☐ 1:YES

OVLP: It is allowed to overlap the next PR. When overlapping
☒ 0:NO ☐ 1:YES

CMD: End of position command
☒ 00: ABS Absolute Position, CMD=DATA
☐ 01: REL Relative Position, CMD=Current Position+DATA
☐ 10: INC Incremental Position, CMD=Previous CMD+DATA
☐ 11: CAP Capture at high speed, CMD=Captured+DATA

ACC: Time Index of accelerating to rated speed(3000rpm) #0
 DEC: Time Index of decelerating from rated #1
 SPD: Target speed index #0
 DLY: Delay time index #0

P6-3: Position CMD DATA(PUU) 29990

Buttons: Cancel, OK, Write to Servo

4.7. ábra - ASDA-Soft - Parameter Setting Helper a P6 – 02 paraméternél [saját kép]

Ahogy a fenti ábrán [4.7. ábra] is láthatjuk, nem kell a dokumentációt elemezve megkeresni újra és újra a beállítási paraméter értékét; ebben az ablakban be tudunk állítani mindent az adott paraméterrel kapcsolatban, ezek alapján beállítja annak értékét. Az elvártak szerint az adagolás folyamata úgy alakul, hogy első lépésként a szervo visszahúzza a fóliát egy rövid (~5-10 mm - es), beállított hosszra, majd leadagolja a beállított adagolási hosszra a fóliát. Ezután következik

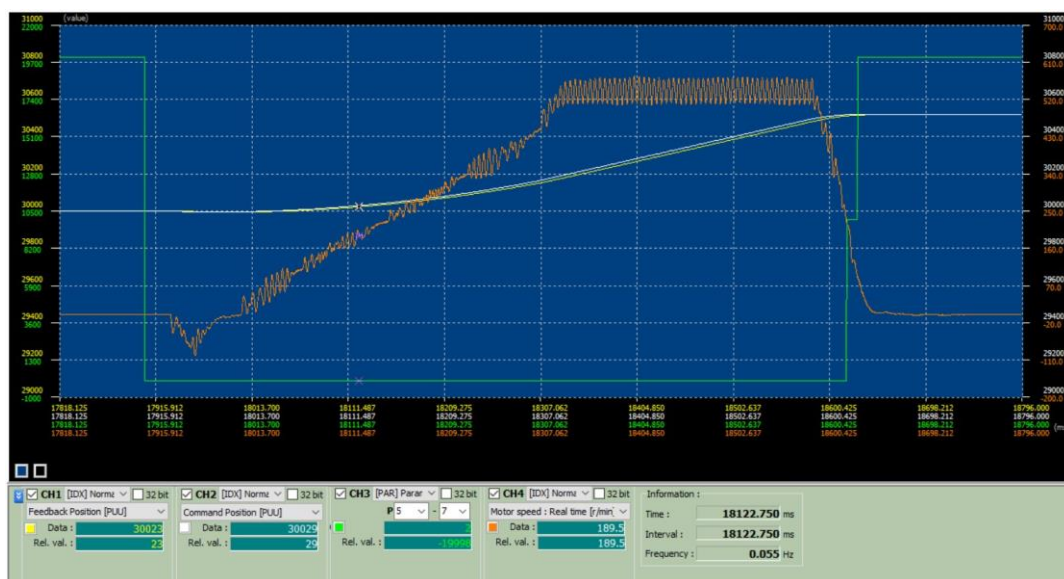
a hegesztés. A visszahúzásra azért van szükség, mert egyes anyagok könnyen leragadhatnak a meleg hegesztőfejre.

Az első próbálkozásokkor mindkét pozíciót Single positioning control-ra állítottam, ami azt jelenti, hogy amint a motor elérte a beállított pozíciót, megáll. Ekkor, az adagolás során a következő lépések zajlottak le:

1. Visszahúzás parancskiadás a szervó felé.
2. Anyag visszahúzása.
3. Motor pozíció kiolvasása a szervóból. (itt nem a P5-07 -es paraméter értékét olvasom vissza, hanem a P5-16 -ost, ami a motor aktuális pozícióját mutatja)
4. Amint PLC -ből azt látom, hogy a motor elérte a visszahúzási pozíciót, az adagolás pozíció triggerelése.
5. Motor pozíció kiolvasása a szervóból.
6. Miután megbizonyosodtam róla, hogy a motor elérte a kívánt pozíciót, kezdődik a hegesztés.

Ezzel a programmal tesztelve a gépet, a vevő nehezményezte, hogy túl sok idő telik el a visszahúzás és az adagolás között, nem indul egyből. Ráadásul nem volt mindig egyforma ez az idő, ami eltelt a két munkafázis között. A hiba kiküszöbölését, mivel a kommunikáció átalakítása továbbra sem vezetett megoldásra, a szervó beállítások módosításával értem el.

A szervó működésével kapcsolatos hibakeresésekben sokat segített az ASDA-Soft „Scope” funkciója, amiben 4 csatornán gyakorlatilag bármi, a szervóban lévő adatot vizsgálhatunk.



4.8. ábra – AZ ASDA-Soft Scope funkciója [saját kép]

Ezt használtam akkor is, mikor az RS-485 -ön keresztüli adatkésés okát próbáltam megtalálni, illetve a most ismertetett hibajelentések okának megismerésében is segített.

Az első, visszahúzásra vonatkozó pozíciót úgy állítottam be, hogy az amint lefut, triggerelje a második pozíciót. Így egyszerűen kihagytam a folyamatból a visszahúzás és az adagolás közti kommunikációs lépéseket. Ez a megoldás sem tökéletes, ugyanis amikor kipróbáltam, és többféle beállítással is teszteltem, nem derült ki pontosan, hogy mikor vált az egyes pozíciók között. Illetve azt vettem észre, hogy nem éri el teljesen a visszahúzás pozíciót, mielőtt adagolni kezdene. Ez a későbbiekben, a markerjel figyelés programozásánál [4.3.4 *Kiegészítő funkció, markerjelfigyelés beépítése.*] Ennek ellenére ez lett a végső megoldás, mivel a megrendelőnek megfelelő volt így a gép működése.

A folyamat legelején, valamint az adagolás és a hegesztés közt a beállított várakozási idő miatt nem annyira jelentős ez a késés. Azonban ez esetben is igyekeztem megoldást találni. Végül úgy módosítottam, hogy néhány digitális ki és bemenetet bekötöttem a villanszerelő segítségével. Nevezetesen a Motor is at zero speed jelet a szervo felől a PLC -be, az Internal position command selection 1~64 Bit0, hogy választhassak a 0. (Home pozíció), vagy a 1. számú pozíció közül, a Command triggered, hogy indíthassam ezen a csatornán a folyamatot. Ezek mellett még a Motor STOP jelet kötöttem be, hogy biztonságosabban megállíthassam a motort probléma esetén. Ezekkel az új jelekkel és átalakításokkal az adagolás lépései a következő képpen alakultak:

1. Visszahúzás parancskiadás a szervo felé. Már digitális jelként.
2. Anyag visszahúzása. Ez a lépés aktiválja az adagolást.
3. Anyag leadagolása.
4. Amikor a szervo kiadta a „Motor is at zero speed” jelet, kezdődhet a hegesztés.

A gép továbbra sem működött megfelelően. A látható hiba, hogy az adagolás után nem indul egyből a hegesztés folyamata, és az adagoláshoz rendelt kifújás is, látszólag tovább tart, mint az adagolás. Ennek oka, hogy ugyan látszólag már megállt a motor, viszont pár milliméteren belül még finomhangolt. Így, bár szemmel nem látszott a mozgása, még nem volt a sebessége 0, ezáltal a várt jel is később érkezett. A várt jelet ekkor kicseréltem a szervo PR command completed jelére. Ennek a jelnek a kiadása hamarabb történik, így ezzel is csökken a ciklusidő. Ezzel az alap gép működése megfelelővé vált, a megrendelő elfogadta.

4.3.4 Kiegészítő funkció, markerjelfigyelés beépítése.

A megrendelő szükségesnek érezte felszerelni a gépet markerjel figyelés funkcióval is. Mint arról korábban szó volt a berendezésen már kialakításra került az ehhez tartozó szenzor helye, illetve a PLC -n is meg volt már a bemeneti jel hozzá. Elméletben a program is átalakítható lenne egy apró módosítással, hogy kezelni tudja ezt az üzemmódot is. Fizikailag a gép egy optikai szenzorral lett kibővítvé, ami képes jelezni, ha a fólia alapszínétől eltérő nyomatot, marker jelet észlel.

Az alap elvárás, hogy be lehessen állítani egy hosszt, ameddig a markerjel figyelés nem aktív. Erre azért van szükség, hogy ha a fólián egyéb nyomat is, vagy a fólia gyűrődése/megcsillanása miatt ne szakadjon félbe az adagolás. Miután ezt a hosszt elértük, érvénybe kell lépnie a markerjel figyelésnek. Tehát amint jelez a szenzor, az adagolást meg kell állítani és a ciklust a hegesztéssel folytatni.

A markerjel figyelő szenzor jelét a PLC CPU moduljának legelső bemenetére kötöttem, mivel ez a bemenet beállítható gyors reagálású – úgy nevezett high speed counter – bemenetnek. ez a gyakorlatban annyit jelent, hogy az így beállított bemeneten megjelenő jel azonnal megjelenik a programban, nem csak a következő ciklus elején, a beolvasási fázisban. Erre a minél gyorsabb reagálás miatt és a jel rövid időtartama miatt volt szükség. Ezt a fajta bemenetet eredetileg enkóderrel való pozíciókövetéshez ajánlja a gyártó.

A legegyszerűbb megoldás az lett volna, ha a programból tudjuk vezérelni a folyamatot. A markerjel érzékelő jele a PLC -be érkezik az adja tovább a hajtás Motor STOP bemenetére. Ehhez a lépésláncot kellene kibővíteni úgy, hogy markerjelfigyeléses üzemmódban az adagolás lépés végén, mikor megérkezik a szervo PR position completed jel, nem a hegesztési lépés következik, hanem egy köztes lépés. Ezalatt a szervo egy meghatározott sebességgel adagol. Szerencsére a szervóban üzemmódváltás nélkül van lehetőségünk ilyen beállítani. Gyakorlatilag ezt is egy pozícióként kezeljük, de a beállításai alapján nem helyet adunk meg hozzá, hanem sebességet. Azért, hogy a fix sebességgel történő adagolás előtt a folyamatban ne legyen egy pillanatnyi szünet, – a korábban ismertetett – a visszahúzás és az adagolás pozíciók közt alkalmazott módon próbáltam beállítani a pozíciókat. Tehát, hogy az adagolás triggerelje a fix sebességűt. Ezt kipróbálva azt tapasztaltam, hogy a lépésláncom elakad abban a lépésben, ahol a PR command completed jelet várjuk. Ennek oka, hogy a szervo továbblép a következő pozícióra, mielőtt az előtte lévő ténylegesen befejezné. Emiatt nem adja ki a várt jelet. Persze a legegyszerűbb és legkézenfekvőbb megoldás az lenne, ha az RS-485 -ön keresztül kiolvasva,

az aktuális pozíció értéké követve tudnánk aktiválni a markerjelfigyelést. Mivel ezt továbbra sem tudtam megtenni valós időben, az akár 3000-mm/s fóliasebesség mellett pedig még egy minimális késés is jelentős túladagolást eredményezhet. Igyekeztem olyan digitális kimenetet találni, amit figyelve tudhatnánk, hogy elértül a markerjel figyelés aktiválásához szükséges hosszt. A techsupport bevonásával sem találtunk megfelelőt. A mentorom azt javasolta, hogy ha nem lehet kiolvasni az aktuális pozíciót, akkor számoljam azt. Először azt mondta, elég lesz a felfutási időket kihagyni a képletből és úgy venni, mintha folyamatos sebességgel haladna a fólia. Ez egyszerű lett volna, de hamar kiderült, hogy a számítás és a valós érték nagyban eltér. Ekkor készítettem egy összetettebb képletet.

```

7. //Markerjelfigyelés előtti hossz
8. "Ido_markerjelig_DB".gyorsulas := (6650 / ("Recept".Adagolás_felfutas / 1000));
9.
10. //Visszahúzás
11. "Ido_markerjelig_DB".vissza_ido := SQRT((2 * "Recept".Visszahuzas_hossz_mm) /
    "Ido_markerjelig_DB".gyorsulas);
12.
13. //Adagolás gyorsulás max hossz
14. "Ido_markerjelig_DB".gyorsulas_ido_max := (("Recept".Adagolás_felfutas / 1000) * (6650 /
    "Recept"."Adagolas_sebesseg_mm/s"));
15. "Ido_markerjelig_DB".gyorsulas_hossz_max := ("Recept"."Adagolas_sebesseg_mm/s" *
    "Ido_markerjelig_DB".gyorsulas_ido_max);
16.
17. //Adagolás + gyorsulás
18. "Ido_markerjelig_DB".teljes_hossz := "Recept".Visszahuzas_hossz_mm +
    "Recept".Adagolas_hossz_mm;
19.
20. IF ("Ido_markerjelig_DB".gyorsulas_hossz_max >= "Ido_markerjelig_DB".teljes_hossz) THEN
21. "Ido_markerjelig_DB".gyorsulas_ido := SQRT((2 * "Ido_markerjelig_DB".teljes_hossz) /
    "Ido_markerjelig_DB".gyorsulas);
22. "Ido_markerjelig_DB".Elore_ido := 0;
23. ELSE
24. "Ido_markerjelig_DB".elore_hossz := "Ido_markerjelig_DB".teljes_hossz -
    "Ido_markerjelig_DB".gyorsulas_hossz_max;
25. "Ido_markerjelig_DB".Elore_ido := ("Ido_markerjelig_DB".elore_hossz /
    "Recept"."Adagolas_sebesseg_mm/s");
26. END_IF;
27.
28. "Konfekcionalas_DB".Ido_markerjelig := LREAL_TO_DINT((( "Ido_markerjelig_DB".vissza_ido +
    "Ido_markerjelig_DB".gyorsulas_ido + "Ido_markerjelig_DB".Elore_ido)*1000)-8);
29.

```

4.5. kód – Részlet a „Sebessegek” funkcióból [saját program]

Ebben a kódrészletben a beállított paraméterek alapján számolom ki, hogy mennyi idő kell a markerjel figyelés előtti hossz leadagolására. Habár igyekeztem a lehető legtöbb adatot figyelembe venni, ez a megoldás sem bizonyult megfelelően pontosnak. Sok próbálkozás után ismét a kommunikációs sebesség növelésével folytattam. Az ekkori próbálkozások alapja a ciklusidő lecsökkentése volt. Ehhez létrehoztam még egy Organization Block -ot. Először a

teljes kommunikációt átraktam ide, ám így még nem tudtam eléggé lecsökkenteni a ciklusidőt, továbbra is késés volt tapasztalható, illetve bizonytalanság a kommunikációban. Ezt követően úgy alakítottam át a rendszert, hogy az alap kommunikáció maradjon az OB1 ben, viszont az OB30 -ban létrehozok egy csak azaktuális pozíciót lekérdező kódot. Markerjel figyelés üzemmódban, mikor kritikus a sebesség, Az OB1ben letiltom a kommunikációt és csak ekkor engedélyezem a rövid lefutású, ám csak a pozícióra vonatkozó, az OB30-ban elhelyezett sorokat.

Ezzel a megoldással sem sikerült kellően növelni a pozíció kiolvasásának pontosságát a kritikus időtartamra.

Ennek az időnek az oka, hogy az RS-485 soros kommunikációnk jellegéből adódóan sorban halad végig a küldeni és olvasni kívánt adatokon, így ciklusonként egy adatot egyszer tudunk kiolvasni a hajtásból. Emellett a szervo oldaláról a kommunikációs folyamatok és a vezérlés időmegosztással fut egymás mellett, mikor nagyobb hangsúly helyeződik a vezérlésre (pl.: adagoláskor) kevesebb erőforrás jut a kommunikációra. Tehát akkor, amikor mi kíváncsiak lennénk az aktuális pozícióra. Erre a problémára sem a Siemens, sem a szervo itthoni képviselője nem tudott megoldást javasolni. Mindkét helyen azt a visszajelzést kaptam, hogy máshol nem szokott ilyen probléma előfordulni, de a beállításaimat átnézve nem találtak egyik oldalon sem olyat, aminek a változtatása megoldást hozott volna. Ezért végső megoldásként a markerjel figyelés aktiválását, egy a receptkezelőből állítható hosszú időzítővel adjuk ki. Így minden termékhez be lehet állítani a megfelelő időtartamot.

Azért, hogy a markerjel figyelőről a szervoba tartó jelen se legyen késés a PLC -ben történő jelfeldolgozás miatt, végül beépítettünk egy gyors működésű relét. Ennek a meghúzását a PLC -ből kiadott „Markerjel_figyeles_be” jel végzi. Ezt a jelet egy, az előbb említett, rövid ciklusidejű OB30 -ban helyeztem el, hossza a receptkezelőből állítható. A markerjelfigyelő szenzor jele pedig ezen a relén van keresztül vezetve a szervo felé.

Ekkorra már csak egy hiba adódott a funkció működésében. A szervo a Motor STOP jelre meglehetősen kiszámíthatatlanul lassult le, nem konzekvensen ugyanúgy. Az erre a jelre való megállásnak nem lehet változtatni a paramétereit, lefutási idejét. Erre a problémára a Delta hazai tech supportja javasolt megoldást. Lehetőség van bizonyos bemenetekre event trigger-t beállítani. Ez a gyakorlatban azt jelenti, hogy amikor az adott bemeneten beállítástól függően fel-, vagy lefutó élt detektál a szervo, azzal egy előre meghatározott PR pozíciót indít el. Erre a bemenetre kötöttük be a Markerjel érzékelőből, a PLC -n keresztül érkező jelet. Az elindított

pozíciót pedig kis értékű, relatív pozíciónak állítottam be. Ezzel biztosítva, hogy az adagolása jel beérkezése után meghatározott lefutással álljon meg. Így sikerült kiküszöbölni a leadagolt hosszokban tapasztalt szórást.

Ezzel ez a funkció is elfogadhatóvá vált a vevő részéről.

5 ÖSSZEFOGLALÁS

5.1 Magyar

A szakdolgozatom témáját a konfekcionáló gép tervezése és programozása adta, amelyet a szakmai gyakorlatom során szerzett tapasztalatok alapján vittem véghez.

Az első szakaszban részletesen bemutatam a vevő elképzeléseit, a gép felépítését és szerkezeti egységeit, ezáltal átfogó képet adva az előttem álló feladatról. A későbbiekben a tervezés folyamatáról, illetve a programozási előkészületekről számoltam be. Ezen munkafolyamatokat az AutoCAD és a TIA Portal programok segítségével végeztem el. Munkám azon részeinél, amelyeknél indokolt volt, a folyamatokat ábrákkal illusztráltam, továbbá bizonyos program részeket a jobb érthetőség érdekében magyarázattal egészítettem ki. Legvégül pedig a végső program megírásának folyamatát és nehézségeit mutattam be.

A projekt során felmerülő összes problémát kifejtettem, a lehetséges megoldási kísérleteket és az azokból levont következtetéseket egymás után ismertettem, így levezetve a végső soron helyesnek bizonyuló megoldásokig. Sok esetben a megrendelő részéről érkező kéréseknek sem volt zökkenőmentes eleget tenni. A legnagyobb kihívást azonban a számomra addig ismeretlen működésű szervo megfelelő beállítása jelentette. Ennek az eszköznek a működését sok kísérletezés árán sikerült részletesen megismerni. Azonban ezeket, főleg utólag, remek tanulási lehetőségeknek látom, hiszen több szempontból kellett az adott helyzetet felmérni, és végül megtalálni a legoptimálisabb megoldást. A teljes kivitelezés alatt, a már meglévő elméleti tudásomat is gyarapíthattam, azonban a gyakorlatban is tesztelhettem azt.

5.2 Angol

The topic of my thesis was the design and programming of a confectioning machine, based on the experience I gained during my internship. The first phase was a detailed description of the customer's ideas, the design of the machine and its components, and thus a comprehensive picture of the task at hand. Later I reported on the planning process and programming preparations. These workflows have been carried out using AutoCAD and the TIA Portal.

In parts of my thesis, where appropriate, I have illustrated processes with diagrams and added explanations to certain parts of the programme for easier understanding. In the end, I described the process and the difficulties of writing the final programme.

I have explained all the problems that have arisen during the project, outlined the possible solutions and the conclusions that have been drawn from them, leading to the solutions that have ultimately proved to be the right ones. In many cases, the customer's requirements were not easy to meet.

However, the biggest challenge was setting up a servo that I had never seen before. Understanding the details of how this device works has taken a lot of experimentation. And yet, I see these challenges, especially with the benefit of hindsight, as great learning opportunities, as I had to look at the situation from different angles and ultimately come up with the most optimal solution. During the whole project I was able to build on my existing theoretical knowledge, but also test it in practice.

ÁBRAJEGYZÉK

2.1. ábra – Lecsévélő [saját fotó]	5
2.2. ábra - Lengőtár [saját fotó].....	5
2.3. ábra - Kiegészítő egység [saját fotó]	6
2.4. ábra - Lyukasztó munkahengerek [saját fotó]	7
2.5. ábra - Fólia felfűzése a pozícionáló hengerekre [saját fotó]	7
2.6. ábra - Adagoló egység [saját fotó].....	8
2.7. ábra – HMI [saját fotó].....	9
2.8. ábra - "Kézi terminál" [saját fotó]	9
2.9. ábra - Adagoló kapcsoló [saját fotó]	10
3.1. ábra - Szervohajtás az első tervváltozatban [forrás: A villamos terv első revíziója].....	12
[\\Szakdolgozat_Major_Miklos_XVREQ7_mellekletek.zip\\Villamos	terv\\
2022_012_VT_A1.dwg]	
4.1. ábra - Részlet a "Munkahenger_FC" -ből [saját program].....	17
[\\Szakdolgozat_Major_Miklos_XVREQ7_mellekletek.zip\\Program\\2022_012_P1\\Munkahenger_FC]	
4.2. ábra – Hibatörlés [saját program].....	21
[\\Szakdolgozat_Major_Miklos_XVREQ7_mellekletek.zip\\Program\\2022_012_P1\\Hibakezelés_FC]	
4.3. ábra - Hibatörlés engedély [saját program]	22
[\\Szakdolgozat_Major_Miklos_XVREQ7_mellekletek.zip\\Program\\2022_012_P1\\Hibakezelés_FC]	
4.4. ábra - A 6T1 -Lecsévélő frekvenciaváltó sebességszabályozása [saját terv]	32
[\\Szakdolgozat_Major_Miklos_XVREQ7_mellekletek.zip\\Villamos	terv\\
2022_012_VT_A3.dwg]	
4.5. ábra - Lecsévélő sebesség visszaosztása [saját program].....	34
[\\Szakdolgozat_Major_Miklos_XVREQ7_mellekletek.zip\\Program\\2022_012_P1\\6T1_FC]	
4.6. ábra - A Lecsévélő motor funkció blokkja [saját program].....	35
[\\Szakdolgozat_Major_Miklos_XVREQ7_mellekletek.zip\\Program\\2022_012_P1\\Motor_FB	
]	

4.7. ábra - ASDA-Soft - Parameter Setting Helper a P6 – 02 paraméternél [saját kép]	37
[saját felvétel az ASDA-Soft Soft(V5.5.0.0) menüjéből]	
4.8. ábra – AZ ASDA-Soft Scope funkciója [saját kép].....	38
[saját felvétel az ASDA-Soft Soft(V5.5.0.0) scope funkciójáról]	

TÁBLÁZATJEGYZÉK

4.1. táblázat - A "Munkahenger_FB" -hoz létrehozott DB-k belső felépítése [saját program]...	19
4.2. táblázat - Az „Adagoló felső hegesztőfej” -re vonatkozó részlet a "Munkahenger_DB" -ből [saját program].....	19
4.3. táblázat – Hibalista [saját program].....	20
4.4. táblázat - Figyelmeztetés lista [saját táblázat].....	21
4.5. táblázat - 6T1-Lecsévélő frekvenciaváltó felparaméterezése [saját táblázat].....	34
4.6. táblázat - A szervohajtás általam beállított paraméterei [saját táblázat].....	36

KÓDJEGYZÉK

4.1. kód - „Vezérles_FC” function [saját program].....	16
4.2. kód - "Munkahenger_FB" [saját program].....	18
4.3. kód – Egy lépés a „Konfekcionalas_ciklus_FC” -ből [saját program].....	29
4.4. kód - Munkahengervezérlés a "Konfekcionalas_ciklus_FC" -ből [saját program].....	31
4.5. kód – Részlet a „Sebessegek” funkcióból [saját program].....	41

FELHASZNÁLT IRODALOM

1. DELTA: Delta High Resolution AC Servo Drive for Network Communication Applications ASDA-A2 Series User Manual (2022.03.07.)
[<https://downloadcenter.deltaww.com/en-US/DownloadCenter>]
2. DELTA: THE MOTION FUNCTION OF ASDA-A2 (letöltve: 2023.05.19.) [Keszeg Dezsőtől emailben, mellékelve]
3. Kollár György: Folyamat leírás (2023) [Kollár Györgytől, emailben]
4. PILZ: PNOZ e1p Operating Manual-21242-EN-10 (letöltve: 2022.12.15.)
[<https://www.pilz.com/en-INT/support/downloads#>]
5. LANBAO: Retro reflection-PSF-DC 3-wire (letöltve: 2022.09.09.) [Kollár Györgytől, emailben]
6. Major Miklós: AUTOMATA KONFEKCIÓKONÁLÓ BERENDEZÉS - KEZELÉS ÉS ÜZEMELTETÉS (2023) [saját munka, mellékelve]
7. RKC: Digital Controller REX-C100/C400/C410/C700/C900 INSTRUCTION MANUAL (letöltve: 2022.10.26.)
[<https://www.rkcinst.co.jp/english/downloads/8933/imnzc22e1/>]
8. SCHEIDER: Limit switches XC Standard XCKN2121G11 (letöltve: 2022.10.08.)
[se.com]
9. SIEMENS: S7-1200: Application Examples for High-Speed Counters (HSC) (2016.12.27)
[[https://support.industry.siemens.com/cs/document/109742346/s7-1200-application-examples-for-high-speed-counters-\(hsc\)?dti=0&lc=en-HU](https://support.industry.siemens.com/cs/document/109742346/s7-1200-application-examples-for-high-speed-counters-(hsc)?dti=0&lc=en-HU)]
10. SIEMENS: How do you establish a MODBUS-RTU communication with STEP 7 (TIA Portal) for the SIMATIC S7-1200? (2017.02.24)
[[https://support.industry.siemens.com/cs/document/47756141/how-do-you-establish-a-modbus-rtu-communication-with-step-7-\(tia-portal\)-for-the-simatic-s7-1200-?dti=0&lc=en-HU](https://support.industry.siemens.com/cs/document/47756141/how-do-you-establish-a-modbus-rtu-communication-with-step-7-(tia-portal)-for-the-simatic-s7-1200-?dti=0&lc=en-HU)]
11. SIEMENS: SIMATIC S7 S7-1200 Programmable controller System Manual (letöltve: 2022.07.11.)
[<https://support.industry.siemens.com/>]
12. SIEMENS: SIMATIC HMI HMI devices Basic Panels 2nd Generation Operating Instructions (letöltve: 2022.07.20.)
[<https://support.industry.siemens.com/>]

13. TECO: L510s Manual - V05 (2021.07.29.) [<https://teco.com.sg/download/1510/>]
14. WENGLOR: Operating Instructions WM03PCT2 (letöltve: 2022.06.20.)
[<https://www.wenglor.com/en/Sensors/Photoelectronic-Sensors/Print-Mark-Readers/Print-Mark-Reader/p/WM03PCT2>]

MELLÉKLETEK

1. Alkatrészek gépkönyvei, deltától kapott, ppt segédlet:
[\\Szakdolgozat_Major_Miklos_XVREQ7_mellekletek.zip\\Alkatrész gépkönyvek]
2. Fázis szimmetria számítás:
[\\Szakdolgozat_Major_Miklos_XVREQ7_mellekletek.zip\\Dokumentumok\\Fázis szimmetria.txt]
3. Gépinformációk:
[\\Szakdolgozat_Major_Miklos_XVREQ7_mellekletek.zip\\Dokumentumok\\2022_012_GI.xlsx]
4. Kezelési útmutató:
[\\Szakdolgozat_Major_Miklos_XVREQ7_mellekletek.zip\\gépkönyv\\2022_012_KU.PDF]
5. Szervohajtás paraméterfájlja:
[\\Szakdolgozat_Major_Miklos_XVREQ7_mellekletek.zip\\Program\\2022_012_7T1_A3.par]
6. Tételjegyzék:
[\\Szakdolgozat_Major_Miklos_XVREQ7_mellekletek.zip\\Anyagrendelés\\2022_012_TJ.xlsx]
7. TIA projekt mappája:
[\\Szakdolgozat_Major_Miklos_XVREQ7_mellekletek.zip\\Program\\2022_012_P1]
8. További képek és videók a gépről:
[\\Szakdolgozat_Major_Miklos_XVREQ7_mellekletek.zip\\Képek]
9. Villamos terv első revíziója:
[\\Szakdolgozat_Major_Miklos_XVREQ7_mellekletek.zip\\Villamos terv\\2022_012_VT_A1.dwg]
10. Villamos terv második revíziója:
[\\Szakdolgozat_Major_Miklos_XVREQ7_mellekletek.zip\\Villamos terv\\2022_012_VT_A2.dwg]
11. Villamos terv végső revíziója:
[\\Szakdolgozat_Major_Miklos_XVREQ7_mellekletek.zip\\Villamos terv\\2022_012_VT_A3.dwg]

12. Villamos terv részletei PDF formátumban: Kapcsoló szekrény, „B”, „C” kötődobozok
fűrási és rendezési tervei, áramút terv:

[\\Szakdolgozat_Major_Miklos_XVREQ7_mellekletek.zip\\Villamos terv]