

SZAKDOLGOZAT

Samu Bence

2023



Magyar Agrár- és Élettudományi Egyetem
Szent István Campus
Mechatronika Szak

**Palettázó ipari robot programozása és távfelügyeleti
rendszerének tervezése**

Belső konzulens: Mayerné Sárközi Eszter
egyetemi adjunktus

Külső konzulens: Szász Richárd
tervezőmérnök

Készítette: Samu Bence
A0FL0k
nappali tagozat

Intézet/Tanszék: Mechatronika Tanszék

Gödöllő
2023

MŰSZAKI INTÉZET
MECHATRONIKAI MÉRNÖK ALAPSZAK
Gépipari mechatronika specializáció

SZAKDOLGOZAT
feladatlap

Samu Bence (OHD5PO)

részére

A szakdolgozat címe:

Palettázó ipari robot programozása és távfelügyeleti rendszerének tervezése

Feladatkiírás:

Bevezetés, szakirodalom feldolgozás, probléma bemutatás, Palettázó folyamat tervezése: a folyamat ismertetése, a PLC programozása, a robot programozása, távfelügyelet, Tovább lépés lehetőségei, gazdasági számítás, összefoglalás

Közreműködő tanszék: Mechatronika

Külső konzulens: Szász Richárd, tervező mérnök, Robert Bosch Elektronikai Kft.

Belső konzulens: Mayerné Dr. Sárközi Eszter, egyetemi adjunktus, MATE, Műszaki Intézet

Beadási határidő: 2023. november 06.

Gödöllő, 2023. szeptember 04.

Jóváhagyom

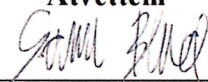


(tanszékvezető)



(szakfelelős)

Átvettem



(hallgató)

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

Gödöllő, 2023. 11. hó 03. nap



(külső konzulens)

Tartalomjegyzék

1. Bevezetés	4
2. Célkitűzés	5
3. Szakirodalmi feldolgozás	6
3.1. Ipari robotok	6
3.2. Programozható vezérlők	12
3.3. Pneumatika	16
4. Palettázó folyamat tervezése	22
4.1. A folyamat ismertetése	23
4.1.1. A paletta és a munkadarab	25
4.1.2. Szállítópálya	26
4.1.3. Pneumatika	27
4.1.1. Szenzorok	31
4.2. A PLC programozása	32
4.2.1. Feladatmegoldás	32
4.2.2. Kontroll panel	35
4.2.3. PLC alap program	37
4.3. A robot programozása	40
4.3.1. A robot felépítése	40
4.3.2. Feladatmegoldás	41
4.3.3. Robotprogram	42
4.4. Távfelügyelet	45
5. Tovább lépési lehetőségek	47
6. Gazdasági számítás	48
7. Összegzés	49
8. Summary	50
9. Felhasznált irodalom	51

1. Bevezetés

Az ipari robotok szerepe és jelentősége a gyártás és automatizálás területén az elmúlt évtizedekben rohamosan növekedett. Az egyre növekvő igények miatt, hogy gyorsabb és pontosabb gyártási folyamatokat hozzanak létre, az ipari robotok váltak a modern gyártás meghatározó eszközeivé. Ezen rendkívüli technológiai fejlődés mögött az áll, hogy az ipari robotok kiválóan alkalmazkodnak a legkülönbözőbb feladatokhoz, az egyszerű rakodástól a bonyolult összeszerelésig.

A palettázás, azaz a munkadarabok vagy termékek elhelyezése és átrakása egyik helyről egy másikra, az ipari robotok kiemelkedő alkalmazási területe közé tartozik. Az ipari robotok képesek hatékonyan és pontosan végrehajtani ezt a feladatot, és lehetővé teszik a gyártási folyamatok magas szintű automatizálását. A palettázási feladatok megkövetelik a precíziót és a gyorsaságot, és az ipari robotok mindezekben a területeken kiválóan teljesítenek.

Ebben a szakdolgozatban egy, az ipari viszonylatban felmerülő palettázási probléma kerül megoldásra egy ipari robot alkalmazásával, valamint a folyamat felügyelése egy weboldalon keresztül. Szó esik majd a folyamat működéséhez szükséges kiegészítő elemekről, mint például a vákuum rendszerről is.

2. Célkitűzés

A szakdolgozatom témája egy palettázó ipari robot programozása és a távfelügyelet megtervezése. A dolgozat célja egy, az ipari kontextusban felmerülő anyagmozgatási probléma megoldása egy ipari robot kar alkalmazásával, valamint képessé tenni a rendszert a pillanatnyi állapotáról való tájékoztatásra az üzemeltető felé.

A projekthez a Magyar Agrár- és Élettudományi Egyetem Műszaki Intézet Mechatronikai Tanszéke biztosítja számomra az alkalmazott robotkart és a PLC-t. A feladat során el fogom készíteni a robot programját a palettázás végrehajtására, valamint a PLC programot amely vezérli a robot munkaciklusát, kezeli a rendszer szenzor jeleit és információt szolgáltat a távfelügyeletnek. Ki fogom választani az anyagmozgatáshoz szükséges szállítópályát, a folyamat irányításhoz szükséges szenzorokat és robotkarra szerelt vákuum megfogó korongot a munkadarabok felvételéhez. El fogom készíteni a HTML alapú távfelügyeleti weboldalt, amely információval szolgál a palettázási ciklus jelenlegi állapotáról és nyomon követi az átpakolt munkadarabok számát, valamint a folyamatban felmerülő hibákról is információt ad majd.

3. Szakirodalmi feldolgozás

3.1. Ipari robotok

Az elmúlt évek során az ipari szektorban nőtt az igény olyan gépek és berendezések iránt, amelyek képesek a termelést gyorsítani és automatizálni, valamint hatékonyan végezni akár komplex feladatokat. A robotika területén zajló fejlesztések ezen igények kielégítését célozzák meg, és az univerzális robotok fejlesztésére összpontosítanak. Ezek a modern ipari robotok gyorsabbak és pontosabbak, mint korábbi modelljeik. [6]

Ezen a ponton felmerül, hogy mit is nevezünk ipari robotnak. Az ISO 8373 szabvány értelmezésében az ipari robot egy automata irányítású, többcélú, újra programozható, három vagy több tengelyű, fix telepítésű vagy mobil, valamint ipari felhasználásra szánt manipulátor. [5]



3.1.ábra: Ipari robotok [20]

A mai ipari robotok összetett mechanikai és elektronikai rendszerek, és számos különböző ipari folyamatban használják őket az ember munkaerő helyettesítésére. Az ipari robotok fejlesztésében és alkalmazásában a korábban külön működő műszaki tudományágak összefonódtak. Ennek következtében a robotika egy új, interdiszciplináris tudományággá vált, amelyben a mechanika, hidraulika, pneumatika, elektrotechnika, irányítástechnika, elektronika, számítástechnika és más műszaki területek egymással összefonódnak.

Az ipari robotok tervezésével, gyártásával, üzembeállításával, alkalmazásával, programozásával és vizsgálatával foglalkozó szakembereknek széles körű ismeretekkel kell rendelkezniük az említett műszaki területek között. Az ipari robotika területén tevékenykedő szakembereknek az interdiszciplináris megközelítés és a széleskörű tudás segít abban, hogy hatékonyan megértsék, tervezzenek és üzemeltessenek modern ipari robotokat. Az ipari robotika egyre növekvő terület, amely kulcsszerepet játszik a gyártás és az automatizálás fejlődésében. [9]

Az ipari robotoknak számos fontos részegysége van, amelyek kulcsfontosságú szerepet játszanak működésükben: [6]

- **Mechanikai váz:** A mechanikai váz a robot testének strukturális alapja, amely lehetővé teszi a mozgást és ellenáll az üzemelés közben fellépő erőknek. Ez a váz hasonló a szervezetünk csontvázához és lehetővé teszi a különböző mozgásokat.
- **Csuklók:** A csuklók a robotkarokban található ízületek, amelyek lehetővé teszik az egyes karvégtagok vagy elemek egymáshoz viszonyított elmozdulását. Ezek nélkül a robot nem lenne képes különböző pozíciókba mozogni.
- **Meghajtó egységek (Aktuátorok):** Az aktuátorok, vagy meghajtó egységek, felelősek a csuklókat mozgató motorok vagy hajtóművek irányításáért. Ezek meghajtók lehetnek hidraulikus, pneumatikus vagy elektromos meghajtásúak, attól függően, hogy milyen feladatokat kell ellátniuk.
- **Irányító egység:** Az irányító egység a robot mozgásának vezérléséért és koordinálásáért felelős. Hasonlóan a központi idegrendszerhez az emberi testben, az irányító egység irányítja a robot mozgását a kívánt célok elérése érdekében.
- **Szenzorok (Érzékelők):** Az érzékelők segítik a robotokat abban, hogy információkat gyűjtsenek a környezetükről. Ezek lehetnek kamerák, lézerszenzorok, hangérzékelők és egyéb érzékelők, amelyek segítik az akadályok észlelését, a munkadarabok pozíciójának meghatározását és egyéb fontos információk beszerzését.

Ezen részegységek kombinációja teszi lehetővé az ipari robotoknak, hogy különböző feladatokat hajtsanak végre a gyártásban és az automatizálás területén. A legfejlettebb

robotok további érzékelő rendszerekkel is rendelkeznek, amelyek segítik őket az adatok gyűjtésében és az intelligens döntéshozatalban a munkafolyamatok során. Az ipari robotok rendkívül fontosak a modern gyártásban és automatizálásban, és folyamatos fejlődésük kulcsfontosságú a hatékonyság és a termelékenység növelésében. [6]

Az ipari robotok irányítása és működése alapjában véve mechanikai alapokon nyugszik. Ezek a gépek többek között mechanikai vázzal rendelkeznek, amelyben különböző kartagok és csuklók találhatók, hogy lehetővé tegyék a robot mozgását és pozícióváltását. Az aktuátorok vagy beavatkozó egységek felelősek a kartagok mozgásának irányításáért, és lehetnek hidraulikus, pneumatikus vagy elektromos hajtások, attól függően, hogy a robot milyen környezetben dolgozik és milyen munkafolyamatokat hajt végre. [13]

Az irányító egység, amely a robot szellemi központja, lehetővé teszi a robotmozgások koordinációját és vezérlését. Ez a részegység lehetővé teszi a robot programozását és a munkafolyamatok hatékony végrehajtását. Az ipari robotok egyre fejlettebb irányítóegységekkel rendelkeznek, amelyek lehetővé teszik a bonyolult munkafolyamatok egyszerű kezelését. [9]

Az érzékelő rendszerek kritikusak az ipari robotok számára, mivel lehetővé teszik a külvilágból érkező információk érzékelését és feldolgozását. Ezek a szenzorok lehetővé teszik a robotok számára, hogy észleljék a környezetüket, érintkezzenek az objektumokkal és reagáljanak a környezeti változásokra. Az érzékelő rendszerek különböző típusúak és méretűek lehetnek, és hozzájárulnak a robotok hatékonyságához és pontos működéséhez.

Az ipari robotika területe folyamatos fejlődésen megy keresztül, és az ipari robotok egyre szélesebb körű alkalmazást találnak a modern gyártásban. Ezek a rendszerek hatékonyabban és pontosabban végzik el a munkafolyamatokat, ami hozzájárul a termelékenység növekedéséhez és a gyártási költségek csökkentéséhez. Az ipari robotika területén zajló technológiai fejlődés továbbra is elősegíti az ipari robotok szerepének növekedését a modern gyártásban. [12]

Az ipari robotok mechanikai alapjai és irányítása egyaránt fontos szerepet játszanak a hatékony munkavégzésben és a széles körű alkalmazásban. Az ipari robotok képesek különböző

munkafolyamatok elvégzésére, és a megfelelő irányítás és mechanikai tervezés lehetővé teszi számukra a feladatok precíz és hatékony végrehajtását. Az ipari robotika területe folyamatosan fejlődik, és az új technológiai fejlesztéseknek köszönhetően az ipari robotok egyre sokoldalúbbak és alkalmazhatóbbak lesznek a gyártásban. [6]

Az ipari robotok programozhatósága és az átprogramozhatóság módja kulcsfontosságú szerepet játszik azok működésében és intelligencia szintjükben. Az alábbi három esetet vizsgálva láthatjuk, hogy mennyire változatosak lehetnek az ipari robotok az átprogramozhatóság és intelligencia tekintetében:

- **Önálló programbefolyásolás nélküli ipari robot:** Ebben az esetben a robotok minden egyes munkaciklusnál az aktuális program inicializálásához klaviatúrájukat használják. Ha rendelkezésre áll egy korábban archivált program, azt betölthetik, különben pedig új programot kell írniuk minden munkaciklushoz. Ez a megközelítés korlátozottabb és kevésbé intelligens, mivel a programozás és inicializálás manuális beavatkozást igényel.
- **Programszelekcióval rendelkező ipari robot:** Ezek a robotok képesek a különböző munkaciklusokhoz tartozó programokat külső vagy belső memóriában tárolni. A programokat külső jelek hatására aktivizálhatják, szelektálhatják, és szabadon sorba rendezhetik. A programszelekciós modul lehetővé teszi a programok gyors kiválasztását a megfelelő mozgásciklushoz vagy technológiai feladathoz tartozó programazonosítók segítségével. Ez a megközelítés már intelligensebb, mivel lehetővé teszi a gyors átállást különböző munkaciklusok között.
- **Programadaptációval rendelkező ipari robot:** Ezek a robotok még fejlettebbek és intelligensebbek. Az automatikus azonosítási rendszer, például egy képfeldolgozó rendszer, segítségével adaptálják működésüket a környezetükben bekövetkező változásokhoz. Amikor az új munkadarabok vagy alkatrészek érkeznek, a robot automatikusan azonosítja őket és alkalmazkodik a változásokhoz. Ehhez fejlett irányító szoftvert és érzékelőrendszereket használnak. A programadaptáció növeli a robot intelligencia szintjét és lehetővé teszi a flexibilis működést a változó környezetben.

Ezek a különböző programozhatósági és átprogramozhatósági módszerek mutatják be, hogy az ipari robotok milyen mértékben képesek alkalmazkodni és reagálni a változó

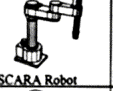
munkakörnyezetre és feladatokra. Az egyre fejlettebb technológiák és szoftverek lehetővé teszik a robotok intelligensebbé válását és hatékonyabb működésüket a modern ipari környezetben. [1]

Az ipari robotokat a kinematikai elrendezés alapján különböző típusokba sorolhatjuk, amelyek meghatározzák mozgásuk módját és alkalmazhatóságukat. Az általános jelöléseket alkalmazva (R-forgó mozgás, T-egyenes vonalú mozgás) a következő elrendezéseket különböztethetjük meg: RRR, RRT, RTR, TRR, TTR, TRT, RTT, TTT. Ezek közül az alábbi felsorolás kiemeli a legelterjedtebbeket: [9]

- **Derékszögű koordinátarendszerű (TTT):** Ezek a robotok három translációs kényszerrel rendelkeznek, amelyek ortogonális módon helyezkednek el egymáshoz képest, létrehozva egy derékszögű koordinátarendszert. Egy ilyen rendszer sokféle általános feladat elvégzésére alkalmas.
- **Henger koordinátarendszerű (RTT):** Ebben az elrendezésben egy forgási kényszer egy translációs kényszer követ. Ezek a robotok olyan alkalmazásokhoz használhatók, ahol a forgás fontos szerepet játszik, például a gépgyártásban.
- **Gömbi koordinátarendszerű (RRT):** Ebben az elrendezésben két forgási kényszer egy translációs kényszer követ. Ezek a robotok a háromdimenziós térben szabadabban mozognak és képesek bonyolultabb feladatok végrehajtására.
- **Csuklós rendszerű (RRR):** A csuklós robotok két fő típusba sorolhatók, a függőleges síkú csuklókaros robotokba és a vízszintes síkú csuklókaros robotokba. Ezek a robotok általában pontosságot és precíziót igénylő alkalmazásokhoz használatosak, például az elektronikai iparban vagy a gyógyszeriparban.

Ezen elrendezések alapján az ipari robotok különböző osztályokba sorolhatók. Emellett innovatív koncepciók, mint a párhuzamos és a trianguláris koncepció is megjelentek a TTT osztállyal kapcsolatban. Ezek a koncepciók lehetővé teszik a robotok számára további speciális mozgási és alkalmazási lehetőségek kifejlesztését. Az ipari robotok változatos kinematikai

elrendezéseik révén különböző feladatokra optimalizálhatók, és a modern gyártásban és automatizálásban fontos szerepet játszanak. [1]

Principle	Kinematic Structure	Workspace
 Cartesian Robot		
 Cylindrical Robot		
 Spherical Robot		
 SCARA Robot		
 Articulated Robot		

3.2.ábra: Robot típusok [21]

A legegyszerűbb ipari robotfeladatok közé tartoznak a rakodási és palettázási feladatok, amelyek gyakran ismertek a "pick and place" feladatokként. Ebben a fajta feladatban a robot fő célja az, hogy alkotórészeket vagy munkadarabokat egy kiindulási helyről egy célpont helyre átrakjon. Általában ebben az esetben csak a kiindulási és a célpont pozíciójának definiálása szükséges, és a robotnak nincs jelentősége a két pont közötti út vagy az átrakott tárgy orientációjának. [12]

Ez a feladattípus gyakran előfordul például a gyártósorokon, ahol a robotnak egyszerűen csak fel kell vennie egy munkadarabot egy szállítószalagról vagy egy rakodási területről, majd el kell helyeznie egy másik helyen, például egy palettán vagy egy dobozban. Az orientáció a legtöbb esetben nem kritikus, és a robot egyszerűen csak a tárgyat veszi fel és helyezi le, anélkül hogy komplex mozgáspályákat kellene kiszámolnia. [13]

Ez a fajta feladat tipikusan a robotok alapvető felhasználási területei közé tartozik, mivel viszonylag egyszerű megoldást kínál a tervezés és a programozás szempontjából. Az ilyen feladatokra tervezett robotok rendszerint egyszerű megfogókkal vagy robotkarokkal rendelkeznek, amelyek könnyen alkalmazkodnak a munkadarabok méretéhez és alakjához. [6]

3.2. Programozható vezérlők

A korszerű iparban az irányítástechnika egyik legfontosabb eleme a programozható logikai vezérlő. Szakirodalmakban sokszor az angol Programmable Logic Controller elnevezés rövidítéseként PLC-nek nevezik. Megeshet, hogy más nyelveken íródott irodalmakban eltérő rövidítést használnak, például a német nyelvben az SPS (Speicher-Programmeirbar Steuerung) jelenik meg. [3]

Az első PLC-ket a General Motor cég megbízására fejlesztették ki, egy 1968-ban hirdetett pályázatra. A pályázat specifikációjában kikötötték a kis méretet, moduláris felépítést, a könnyű újra programozhatóságot, a valós idejű működést és a megbízhatóságot. A pályázatot két cég, a Modicon és az Allen-Bradley teljesítette, amelyek később vezető gyártókká váltak a PLC piacon. [7]

A tényleges fejlődés a területen a mikroprocesszorok megjelenésével indult meg, amelyek egyre több I/O csatornát voltak képesek kezelni. Emellett a fejlődésükkel növekedett a program- és az operatív tár kapacitása.

A következő nagy előre lépés a területen az úgynevezett „mini” PLK-k kifejlesztése volt, amelyek olcsó és kompakt megoldást jelentettek az ipari vezérlőrendszerek számára. Elterjedté vált a PLC-k hálózatba kapcsolása, amely lehetőséget nyújtott a számítógépekkel való integrációra. Ez által lehetővé vált a távfelügyelet, valamint a távoli programozás is. [3]



3.3.ábra: Mini PLC [16]

A napjainkban alkalmazott PLC-k két csoportba sorolhatók a kivitelük alapján: kompakt és moduláris. [10]

A *kompakt* PLC-k nevükből is eredően kis helyigényűek, amelyeket megfelelő biztonsági tokozásban készítenek ipari felhasználásra. Hátránya ennek a csoportnak hogy a hardver struktúrája nem módosítható. A kompakt PLC-khez tartoznak a mikro-PLC-k, amelyek tartalmazzák az ember-gép kapcsolathoz szükséges hardver- és szoftverfeltételét is. Felhasználási területükhöz tartozik a sorozat gyártott gépek és berendezések, valamint az egyedi vezérlések.

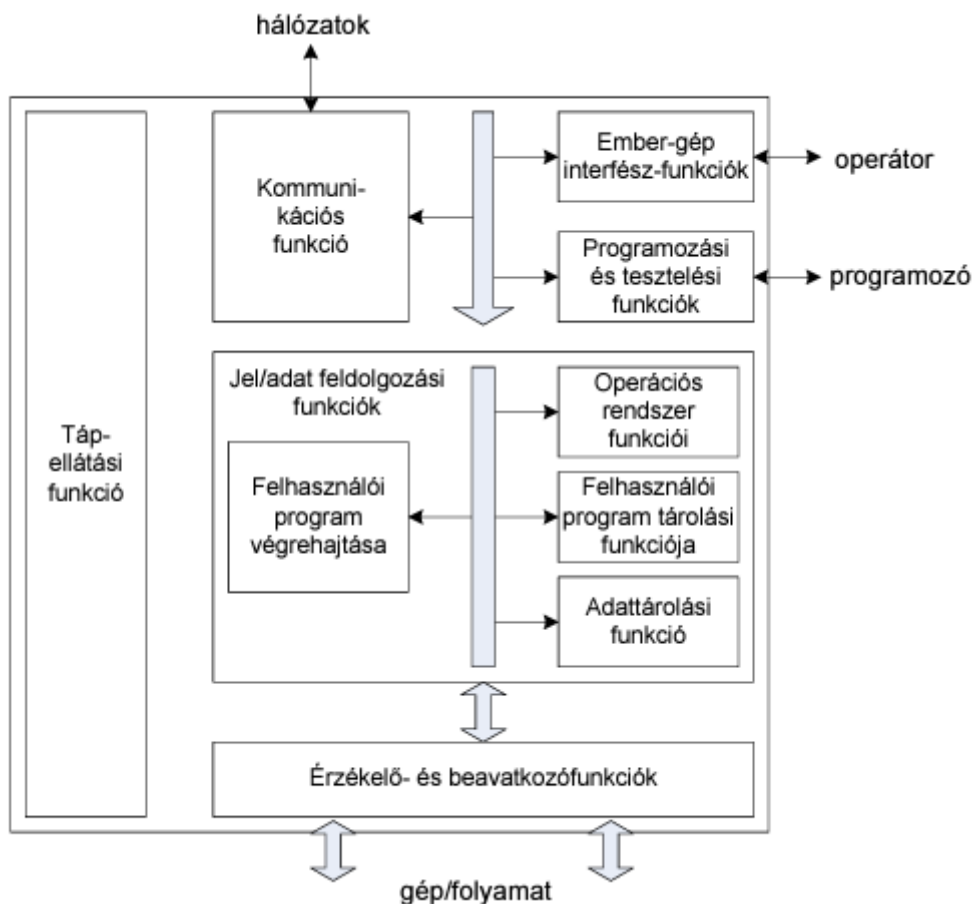
A *moduláris* kialakítású programozható logikai vezérlők konfigurációja módosítható, mivel valamilyen speciális funkciót végrehajtó modulokból állnak. A modulok úgynevezett rack-be dugaszolhatók, műszerdobozba vagy szekrénybe szerelhetők. Ezt a típusú PLC-ket rugalmas gyártórendszerek és ipari folyamatok irányítására hozták létre. [7]

A korszerű moduláris PLC-k vagy rendelkeznek Ethernet interfésszel a hagyományos kommunikációs kapcsolatokon kívül, vagy lehet egy Ethernet modult kapcsolni hozzájuk. [3]

A PLC funkcionális szempontból az alábbi részegységekből tevődnek össze:

- a központi logikai egység
- a programmemória
- az adatmemória
- be- és kimeneti egységek
- kommunikációs csatornák
- tápegység [7]

Ezek a részegységek ellátott funkcióknak az egymáshoz való viszonyulását szemlélteti az alábbi 3.4.ábra. Ezen felül látható az ábrán hogy a PLC melyik funkciója felelős az operátorral, a programozóval, a géppel vagy folyamattal, valamint a hálózatokkal való kommunikációért.



3.4.ábra: A PLC funkciói [15]

A *központi logikai egység* feladata a betöltött programban található utasítások valós időben való végrehajtása. Ennek érdekében a program ciklikus működésű és a másodperc töredéke alatt képes többször is végig futtatni a programot. Az utasítások alapján feldolgozza a beérkező jeleket, majd a megfelelő kimenetre csatornán ad jelet. Ezen felül a központi logikai egység felel a PLC belső funkcióinak működéséhez szükséges vezérlőjelek előállításáért, ez által biztosítva a különböző egységek szinkronizált működését. [10]

A *programmemória* felelős a PLC-re feltöltött programok tárolásáért, ideértve a rendszerprogramot, amely a PLC operációs rendszereként szolgál. Ennek az elkészítéséért a gyártó cég felel, ROM vagy EPROM típusú, valamint csak olvasható. Ezen felül ebben a memóriában kerülnek elhelyezésre a felhasználói programok is. Ahhoz hogy hozzá lehessen

férni ezekhez a programokhoz, biztosítani kell a tár és a programozó közötti kommunikációt, amely azt jelenti hogy a programozó lehet egy erre a célra kifejlesztett eszköz, egy személyi számítógép, vagy valamilyen interfész. A betöltés és tesztelés után a felhasználói program is egy csak olvasható memóriában kerül elhelyezésre, amely lehet FlashROM, MMC vagy EPROM. [11]

Az *adatmemória* kerülnek az utasításokhoz tartozó változók értékei, a be- és kimeneti csatornák memória képei, valamint a program futása alatt többször is felhasznált kimeneti értékek.

A *be- és kimeneti* illesztőkön keresztül csatlakozik a programozható vezérlő a vezérlési elemekhez. A kimeneti csatornákon keresztül kapcsolódnak a folyamat beavatkozó szervei, a bemeneti csatornákra pedig a bemeneti elemek, ideértve a kapcsolókat és érzékelőket. Ezek a csatornák mindegyike jól meghatározott címmel van ellátva. Általában több bemeneti csatorna van egy PLC-n mint kimeneti, mivel a legtöbb vezérlési feladat esetében több bementi jel van mint kimeneti beavatkozó. [10]

A *kommunikációs csatorna* a PLC-k közötti kommunikáció létesítésére szolgál, amely a gyakorlatban magvalósítható Ethernet hálózaton vagy egy terepi buszon keresztül.

A *tápegység* felelős a programozható vezérlő áram ellátásáért, a vezérlés többi elemétől függetlenül, annak érdekében, hogy zavartalanul működhessen az egység. [3]

Összefoglalva a programozható logikai vezérlők több különböző feladat ellátására is képesek:

- jelfeldolgozás
- technológiai interfész (érzékelők és beavatkozók kezelése)
- kommunikáció (PLC-PLC, PLC-számítógép, PLC-hálózat)
- ember és gép közötti interfész
- programozás
- tesztelés
- dokumentálás
- tápellátás [7]

A napjainkban használt PLC-k tekinthetők ipari számítógépeknek is, mivel a technológia fejlődésével egyre inkább megegyeztek a számítógépek és a programozható vezérlők funkciói. Ebben a tekintetben a PLC egy olyan számítógép, amely speciális hardver egységekkel, valamint felhasználói programjai által képes az ipari technológiai folyamatok szabályozására és vezérlésére. Ezen felül az intelligens kommunikációs felület lehetővé teszi a hierarchikus vagy osztott rendszerek megalkotását. [7]

A programozható logikai vezérlő előnyei közül talán a legjelentősebb a felhasználói programozhatósága. Ennek köszönhetően a felhasználó a tárolt egyedi programmal képes az univerzális hardvert alkalmassá tenni különböző feladatokra. Ebből adódóan a programozási és informatikai ismeretek kerültek az előtérbe a kezelők körében, szemben az áramkör tervezői ismeretekkel. [7]

3.3. Pneumatika

Pneumatikának nevezzük a műszaki tudomány azon területét, amely a levegő nyomását használja fel munkavégzésre és információ közvetítésre.

Sokoldalú felhasználási lehetőségei miatt több ága alakult ki a területnek, amelyeket a nyomásszint alapján lehet csoportokba osztani. [8]

Kis nyomású pneumatikát kizárólag vezérlési célokra alkalmaznak, mivel 0,2 bar alatti értékek tartoznak ebbe a csoportba. A normál nyomás 0,2 és 2 bar között van meghatározva, amelyet például irányítási feladatok esetében alkalmaznak. 2 és 10 bar között már nagy nyomású pneumatikáról beszélhetünk, amelyet többek között automatizálásnál és szállításnál használnak. 10 bar feletti nyomású pneumatikus rendszereket úgynevezett „Igen nagy nyomású” rendszernek neveznek. Ilyenek például a repülőgépekben és hajókban fordulnak elő. [4]

A pneumatikus rendszerek viszonylag gyorsan terjedtek el az iparban, amelynek fő oka hogy a sűrített levegővel megvalósíthatók olyan megoldások, amelyek más energiahordozóval nem vagy nem lennének gazdaságosak. [2]

Ha energiahordozók szempontjából vizsgáljuk a sűrített levegőt, akkor az alábbiak a fő előnyök:

- **Elérhetőség:** Levegő gyakorlatilag mindenütt rendelkezésre áll korlátlan mennyiségben.
- **Szállítás:** Sűrített levegőt egyszerűen és könnyen lehet nagy távolságokra szállítani csővezetékeken, és nincs szükség az elhasznált levegő visszavezetésére.
- **Tárolhatóság:** A sűrített levegőt tartályban lehet tárolni, így nem szükséges a kompresszort állandóan üzemeltetni.
- **Hőmérséklettűrés:** A sűrített levegő nem érzékeny a hőmérsékletváltozásokra, így különleges hőmérsékleti körülmények között is biztonságosan használható.
- **Biztonság:** A sűrített levegő robbanás- és tűzbiztos, nem igényel drága biztonsági berendezéseket.
- **Tisztaság:** A sűrített levegő tiszta, és nem szennyezi a vezetékeket vagy az elemeket.
- **Felépítés:** Pneumatikus rendszerek munkavégző elemei egyszerűek és olcsók.
- **Sebesség:** A sűrített levegő magas áramlási sebességet tesz lehetővé, így gyors munkasebességek érhetők el.
- **Állíthatóság:** A sűrített levegőnél a sebesség és az erő fokozatmentesen vezérelhető.
- **Túlterhelhetőség:** Pneumatikus eszközök túlterhelhetők anélkül, hogy meghibásodnának.

A pneumatika hátrányai:

- **Előkészítés:** A sűrített levegő gondos előkészítést igényel, hogy eltávolítsa az energiahordozó szennyeződést és nedvességet.
- **Összenyomhatóság:** A sűrített levegővel működő hengerek nem képesek terhelés-független, egyenletes dugattyúsebességet biztosítani.
- **Erőkifejtés:** A sűrített levegő gazdaságosan csak egy meghatározott erőkifejtésig alkalmazható.
- **Kipufogás:** A kipufogó levegő zajos, de modern hangtompító anyagokkal ez a probléma csökkenthető.

- **Költségek:** A sűrített levegő viszonylag drága energiahordozó, de magas energiaköltségek kompenzálhatók olcsó alkatrészekkel és nagy teljesítménnyel. [8]

Első lépésként szükséges a kültéri levegő sűrítése, vagyis a megfelelő nyomásra emelése. A kompresszorok erre a célra szolgálnak: csökkentik a gáz térfogatát, emelik a nyomását, és emellett növelik a hőmérsékletét is. Ennek eredményeképpen a kompresszorban hő keletkezik, amelynek elvezetése szükséges. A keletkező hő mennyiségétől függően ezt lehet hűtőbordákkal, külső levegőhűtéssel, vagy nagyobb teljesítményű kompresszoroknál vízhűtéssel megvalósítani. A hatékony hűtés nemcsak meghosszabbítja a kompresszor élettartamát, hanem javítja a sűrített levegő hőmérsékletét is. A vízhűtés bevezetésével a levegőhűtés megtakarítható, vagy alacsonyabb teljesítményen is működtethető. [2]



3.5.ábra: Pneumatikus kompresszor [22]

A kompresszorok működésük alapján két fő csoportba sorolhatók:

- **Térfogatkiszorításos kompresszorok**, amelyek esetében a levegő valamilyen zárt térben összenyomódik vagy egyszerűen áramlik a szívó- és kimenőcsők között. Ezek a kompresszorok lehetnek egyenes vonalú mozgást használók, forgattyús mechanizmussal ellátottak vagy forgómozgású berendezések.
- **Áramlásdinamikai elven működő kompresszorok**, amelyek a levegőt egy nyitott térben gyorsítva sűrítik [2]

Térfogatkiszorításosra példa a dugattyús kompresszorok, amelyek működése során egy forgattyús mechanizmus által mozgatott dugattyú hengerben végez alternáló mozgást. A hengerfejen elhelyezkedik egy vagy több szívó- és nyomószelep. Ezeket a szelepeket vagy a gáznyomás különbsége, vagy hasonlóan a gőzgépekhez és dugattyús motorokhoz, vezérlés révén nyitja és zárja.

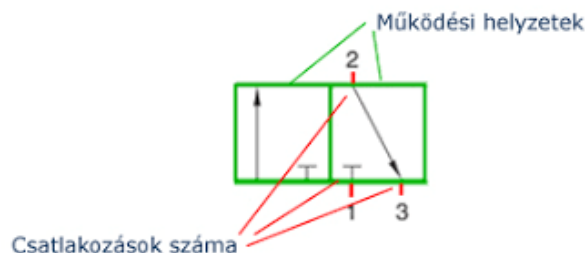
Áramlásdinamikai elven működik az axiális kompresszor, amelyekben a forgórészen elhelyezett szárnyprofil alakú lapátok, amelyeket ellentétesen hajlított, úgynevezett állólapátok követnek. Egy ilyen párost fokozatnak nevezünk. Az axiális kompresszorokban általában több ilyen fokozat található, amelyek növelik a kompressziós viszonyt. Az egymást követő fokozatok keresztmetszete folyamatosan csökken, hogy az axiális áramlási sebesség állandó maradjon. A levegő tengelyirányban lép be a rendszerbe, és onnan is távozik. [2]

A levegő előkészítők felelősek a szennyeződések kiszűréséért, amely lehet szilárd szemcse vagy olaj és vízcseppek. Számos módja létezik ennek a feladatnak, azonban ebben a dolgozatban nincs lehetőség az összes felsorolására.

A pneumatikában szelepeknek nevezzük az irányítási feladatokat ellátó elemeit a rendszernek. Ezeket a szelepeket többféle képen feloszthatjuk, az alábbi felsorolás egy áttekintést nyújt ezek az elemek fő csoportjaira. [4]

- Működtetés szerint
 - elektromos
 - pneumatikus
 - mechanikus
- Működésmód
 - útváltók
 - zárószelepek
 - mennyiségirányítók
 - nyomásirányítók

Az alábbiakban az útváltók kerülnek részletezésre. Az útváltók az áramlás kezdetét, végét és az áramlás irányát határozzák meg, valamint a vezérlőjel hatására változtatják meg a kapcsolási állapotukat úgy, hogy összekössék vagy elválasszák a különböző nyomásszintű területeket. A digitális működésmódot alkalmazzák, amely lehet nyitott (1) vagy zárt (0). Általában egy tört jellel jellemzik az útváltókat, ahol az első szám a szelep csatlakozóinak számát, a második pedig az állások számát jelöli. Példaképpen felhozható a gyakran alkalmazott 3/2-es útváltó szelep, amely két állás között képes váltani és három csatlakozóval rendelkezik. [4]



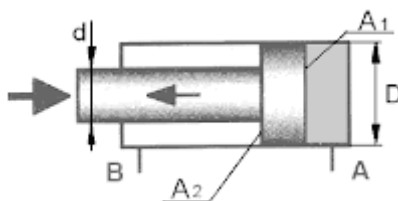
3.6.ábra: 3/2-es útváltó szelep

A kétállású útváltók lehetnek mono- vagy bistabil kiépítésűek, amelyek között a különbség, hogy a monostabil a vezérlő jel megszakadása esetén visszaáll egy meghatározott állapotba. A visszaállási helyzettől függően lehetnek alaphelyzetben nyitott vagy alaphelyzetben zárt útváltókról beszélni. Az útváltók tervezésénél fontos szempontnak kell lennie az átfolyási ellenállás minimalizálása. [4]

A pneumatikus rendszerben lévő energiát aktuatorokon keresztül alakítjuk át mechanikus energiává. Ilyen aktuatorok például a munkahengerek és a légmotorok. [2]

A munkahengerek olyan eszközök, amelyek által a nyomás által kifejtett erőt átalakítják elmozdulássá. A munkahengerek működése során a dugattyú felületére ható közeg nyomása erőt gyakorol, amely a dugattyút mozgásba hozza ($F=p \cdot A$, ahol F az erő, p a nyomás, és A a felület).

A dugattyú mozgásának iránya attól függ, hogy melyik oldalon hat a nagyobb erő. Ebben az esetben a dugattyú az ellenkező irányba mozog, és a kamra tartalmát kiengedi. [2] Az egyoldali dugattyúrúd kivezetéses munkahenger esetén a dugattyú felülete a dugattyúrúd által lefedett terület miatt a dugattyúrúd felőli oldalon kisebb. Tehát ha a két kamra nyomása azonos, a dugattyúrúd kifelé mozog. Ennek a különbségnek az előnye, hogy vannak olyan alkalmazások, ahol csak az egyik irányban fontos az erő (például prések, ahol csak a nyomás kifejtése fontos, és a visszahúzásnak csak annyira kell történnie, hogy a munkadarab emelkedjen). A dugattyúrúd nélküli és az átmenő dugattyúrudas munkahengerek esetén a dugattyú felületére ható nyomás erőssége mindkét irányban azonos, mivel a felület azonos mindkét oldalon.



3.7.ábra: Egyszeres működtetésű munkahenger [17]

Az egyszeres működtetésű munkahengerek hengerterében csak az egyik oldalon található nyomásbemeneti pont, ami azt jelenti, hogy a munkahenger kialakításától függően vagy a rugóoldalon, vagy a dugattyúoldalon végez munkát. A dugattyú nyugalmi helyzetébe állítását általában egy megfelelően méretezett rugó vagy a dugattyú külső terhelése végzi. [2]

Ezek a hengerek általában rövid löketűek, és általában kb. 100 mm lökethosszig használhatók. Ezek a rendszerek olyan munkafolyamatokhoz használhatók, ahol csak egy irányú erő kifejtésre van szükség, és a lökethossz korlátozott. Rugós kialakításuk miatt hatékonyan alkalmazhatók olyan alkalmazásokban, ahol a dugattyút egy rugó vagy külső terhelés nyomja előre vagy húzza hátra a hengerben. Az egyszeres működtetésű munkahengereket gyakran alkalmazzák kilökésre, sajtolásra, emelésre és adagolásra. [2]

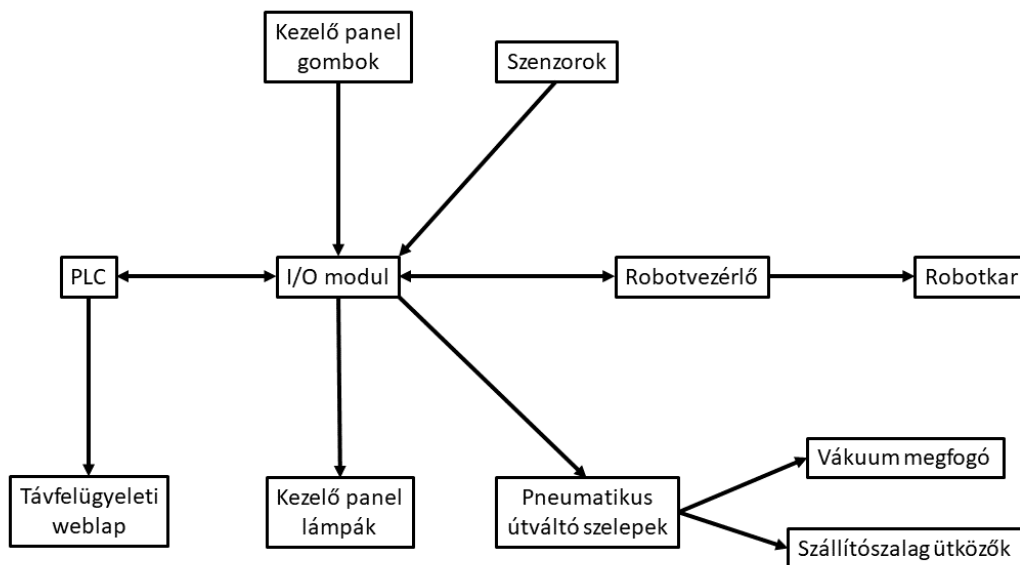
4. Palettázó folyamat tervezése

A szakdolgozat projektje egy szállítószalagról érkező dobozolt termékek egy palettára való átpakolása. A rendszer egy robotkarból, a vezérlő PLC-ből és az alkalmazott szenzorokból és pneumatikus eszközökből áll. A feladat része volt a paletta tervezése, a PLC program megírása, a robotkar programozása, a távfelügyelet megoldása, valamint a szállítópálya, a megfogó és a szenzorok kiválasztása.

A munkadarabokat a szállítópálya juttatja el a robotkar munkaterébe, ahol egy ütköző segítségével kerül a munkadarab a megfelelő pozícióba. Itt egy szenzor került elhelyezésre, amely észleli a munkadarab jelenlétét és jelzi ezt a PLC részére. Innen a munkadarabot egy öt tengelyű robotkar viszi tovább egy vákuum megfogó rendszer segítségével. A célállomás a paletta, ahova a robot elhelyezi a munkadarabot a megfelelő helyre és orientációban. Ennek a palettának a jelenlétét egy másik szenzor ellenőrzi, amely szintén a PLC-nek ad jelet.

A robot mozgását irányító parancsokat a kezelő egy vezérlő panelen keresztül tudja leadni, amelynek jelző lámpái adnak információt a beavatkozás szükségességéről a kezelő felé. A rendszer elemei közötti kommunikáció egy I/O modulon keresztül lett megoldva. A robot és a PLC közt két irányú kommunikáció áll fent, amely információt szolgáltat a rendszer pillanatnyi állapotáról. Ezeket az adatokat a távfelügyeleti weboldal a PLC-ből tud kiolvasni.

Az alábbi 4.1.ábra illusztrálja a rendszer eleminek összeköttetését.

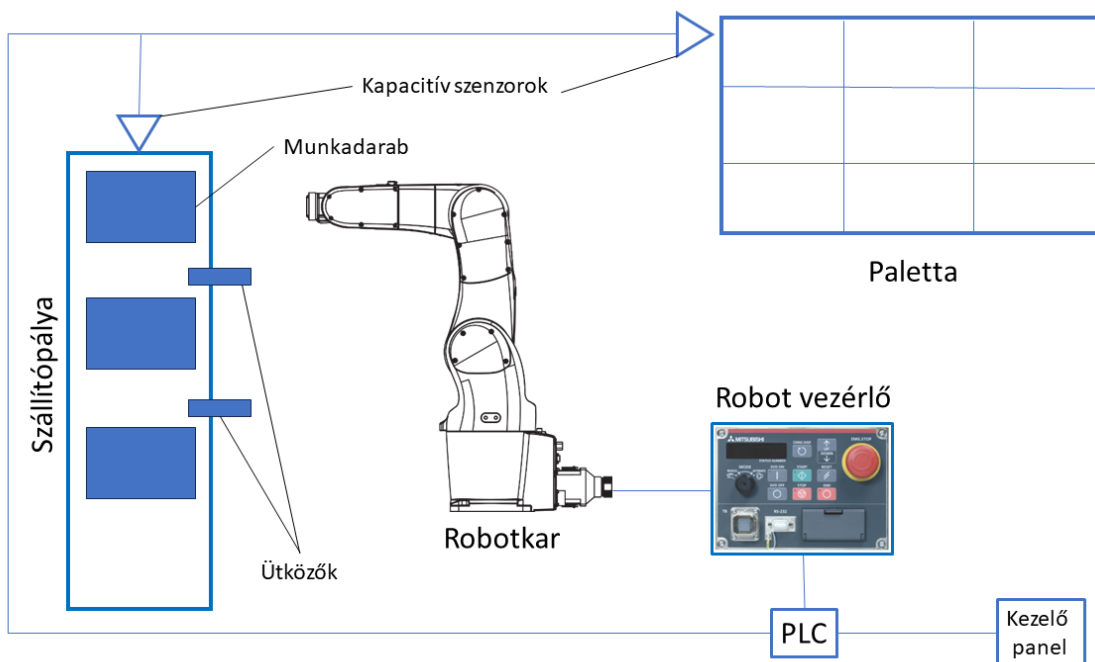


4.1.ábra: Rendszer elemek összeköttetése

4.1. A folyamat ismertetése

Az előzőekben leírtak alapján ebben a rendszerben a PLC a központi elem, ezen keresztül van a többi elem összekapcsolva. A robotkar, a szállítópálya, de még bizonyos értelemben a távfelügyeleti weboldal is csak végrehajtó funkciót látnak el.

A folyamat úgy épül fel, hogy a szállítópályán érkezik a munkadarab, amelynek a megfelelő pozícióba kerülését egy szenzorral ellenőrzi a rendszer. A robotkar ebben az esetben egy vákuum megfogó segítségével kiemeli a terméket, majd a ciklusban betöltött helye alapján elhelyezi a palettán a megfelelő pozícióba. A robot programon belül követi a rendszer hogy hány munkadarabot pakolt már át, amely ha eléri a beállított értéket jelzést ad a PLC-n keresztül a kezelő felé. Ameddig folyik a paletta cseréje a robot a kezdő pozícióba áll vissza és csak akkor indul tovább, ha megbizonyosodott egy új, üres paletta jelenlétéről.



4.2.ábra: A palettázó rendszer felépítése

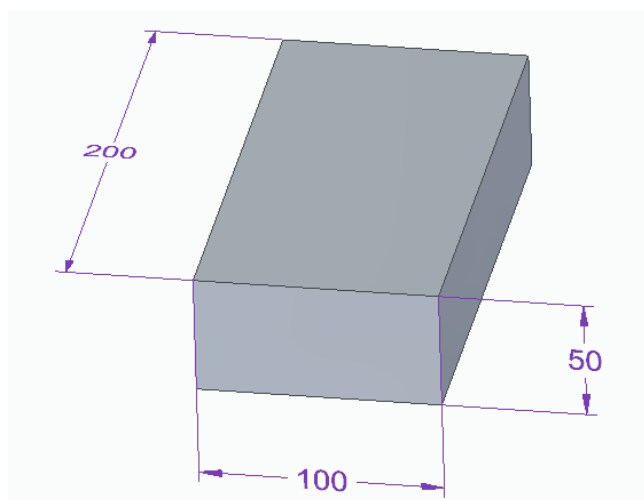
A rendszer rendelkezik egy indítási protokollal, amelynek feladata az egyes elemek üzemképességének ellenőrzése. Ezen felül van lehetőség a folyamat leállítására, vészleállítására, valamint hibakereséssel jelezni a kezelő számára a felmerülő probléma okát. Bár a rendszer képes automata üzemben működni, a jelenlegi kialakításban a paletta megtelése esetén szükség van manuális beavatkozásra, a csere ideje alatt a rendszer várakozó állapotban van.

A PLC és a robot programokat, valamint a távfelügyeleti weboldalt későbbi fejezetekben részletezem. Az alábbi alfejezetekben a rendszerhez tartozó paletta, szállítópálya, megfogó és szenzorok kerültek kidolgozásra, ideértve a kiválasztásukat és jellemzésüket.

A palettázó rendszer elektromos kapcsolásait a mellékletekben található ábra tartalmazza.

4.1.1. A paletta és a munkadarab

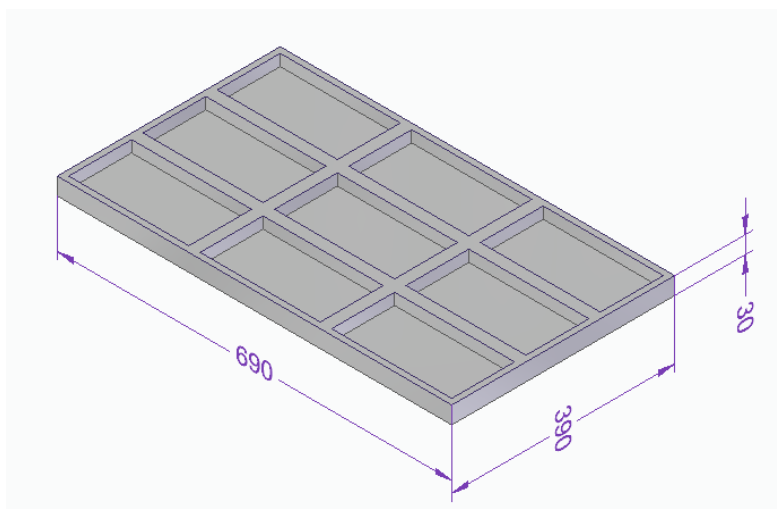
A folyamat során használt paletta egy műanyag tálca, amely 9 darab munkadarabot tud tárolni egy 3x3-as rácsban elhelyezve. A paletta pozícionálását a tartó környezet kialakítása végzi, egyszerű alak tartással biztosítja a paletta rögzítését. A palettára felhelyezett munkadarabok egy karton dobozba csomagolt termékekből állnak, amelyek fő méreteit az alábbi ábra tartalmazza.



4.3.ábra: Munkadarab méretek

A munkadarab egy karton dobozba csomagolt termék, amelyben hungarocell elemek tartják a terméket a megfelelő pozícióban. A munkadarab súlypontja ez által biztosítva van, a 4 vákuum korongok középpontjába esik.

A palettát a munkadarab méreteiből következően határoztam meg, amelyben a 9 lerakható munkadarabnak az elcsúszás elkerülésére megfelelő méretű peremet hoztam létre. A tálca anyaga polipropilén. Az alábbi ábra tartalmazza a paletta fő méreteit.



4.4.ábra: Paletta méretek

A paletta méreteinek definiálása a robot programban kerül sor, ahol a sarok pontok alapján kerül kiszámításra a munkadarabok elhelyezési pontjai. Fontos megemlíteni, hogy ezeknek a pontoknak a felvételénél vigyáznom kellett a robot szerszám orientációjára, mivel ez alapján számítja ki a robot a lehelyezési orientációt a munkadaraboknál.

4.1.2. Szállítópálya

A szállítópálya kiválasztásánál a munkadarab méretein kívül figyelembe kellett vennem a hajtás folyamatosságát is. A munkafolyamat során folyamatosan érkeznek a munkadarabok, ez által felhalmozódnak a szállítópálya végén az ütközők előtt. Ha a szállítószalag ezt az utolsó szekcióját hajtott kivitelűnek választom, akkor felmerül a munkadarabok kopásának problémája is. Ennek elkerülésére ez az utolsó szakasz a pálya vége felé döntött pozícióban kerül beépítésre, ez által az érkező munkadarabok lendülete és a gravitáció juttatja el a munkadarabokat a robotkar felvételi pozíciójába.

A feltételek alapján a Tech-Con D32 görgős pályáját választottam, amelynek szélességét a munkadarab hosszából 200 mm-re állítottam. A munkadarab tömegéből adódóan az 5 kg teherbírású PVC görgőket választottam.

4.1.3. Pneumatika

4.1.3.1. Vákuum rendszer

A vákuum megfogó kiválasztásánál fontos volt a nagy sebességgel való oldal irányú mozgatás közben is biztosítani a fogást. Ennek érdekében első lépésként számítást végeztem a szükséges erőre a mozgás közben, amelyet az alábbi képlet részletez. A biztonsági tényező 4-nek lett megválasztva, a súrlódási együttható pedig 0,6.

$$F_h = \sqrt{F_g^2 + (F_a * \mu)^2} * S = \sqrt{(m * g)^2 + (m * a * \mu)^2} * 4$$

$$F_h = \sqrt{\left(0,5 \text{ kg} * 9,81 \frac{\text{m}}{\text{s}^2}\right)^2 + \left(0,5 \text{ kg} * 9,81 \frac{\text{m}}{\text{s}^2} * 0,6\right)^2} * 4 = 22,88 \text{ N}$$

Természetesen a számítás pontossága érdekében korlátozni kell a robotkar gyorsulását üzem közben $5 \frac{\text{m}}{\text{s}^2}$ -re. A munkadarab stabilitásának érdekében a megfogón 4 darab vákuum korongot helyeztem el, így egy korongnak 5,72 N erőt kell kifejtenie. A korongok szükséges méreteinek számítását az alábbi egyenlet tartalmazza, amelyben 50%-os vákuum van feltételezve.

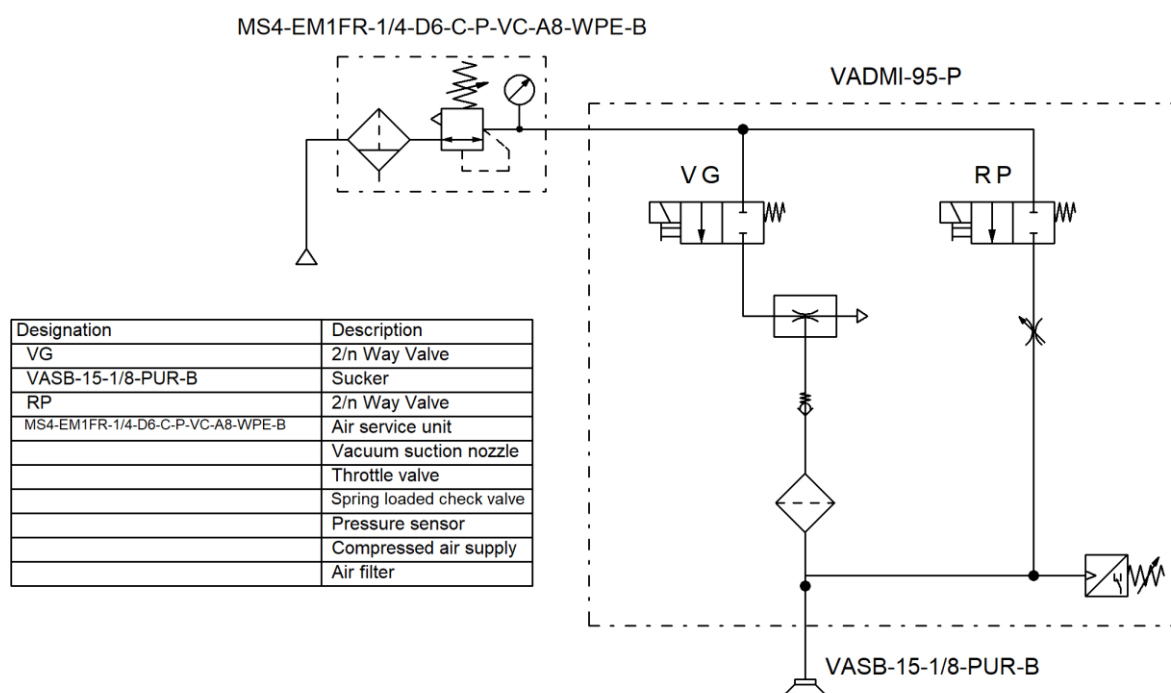
$$d_{min} = \sqrt{\frac{4 * F}{\Delta P * \pi}} = 12,06 \text{ mm}$$

A számított értékek alapján és a munkadarab felületének figyelembevételével a Festo VASB-15-1/8-PUR-B vákuum korongját választottam, amelynek 12,4 mm-es hatékony szívóátmérője eleget tesz a követelményeknek.



4.5.ábra: Vákuumkorong [18]

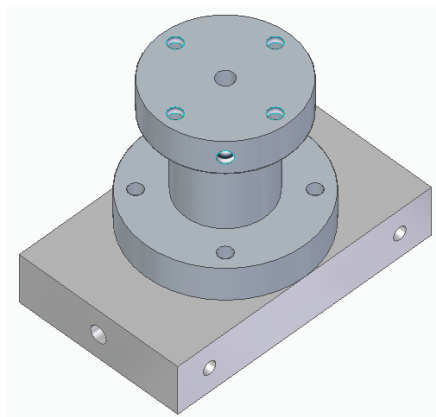
A vákuum generátornak a Festo VADMI-95-P termékét választottam, amelyben egy integrált szelep szabályozza a sűrített levegőellátást a vákuum létrehozására, ez által takarékosabb megoldást biztosítva.



4.6.ábra: Vákuum rendszer

A vákuum megfogó vezérlése a robot programban történik, amely a robotkar elektromos kapcsolásain keresztül vezérli az útváltó szelepet. A hibaellenőrzés a vákuumgenerátorban integrált szenzorokon keresztül történik.

A vákuum korongok rögzítésére és a roboton keresztül átfutó pneumatikus kapcsolás biztosítására terveztem egy két elemből álló tartó szerkezetet. A vákuum az alumínium elembe található furaton keresztül ér el a korongokig, a két elem pedig M5-ös csavarok segítségével van egymáshoz és a robotkarhoz rögzítve. A Mitsubishi RV-2AJ robotkaron lehetőség van az integrált pneumatikus vezetékekre kapcsolni az effektort, amelynek a bevezetése a robot elektromos bekötéseinél található, a kivezetés pedig a negyedik és ötödik csuklók közt kapott helyet. Az alábbi ábra mutatja ennek a tartó szerkezetnek az összeépített modelljét.



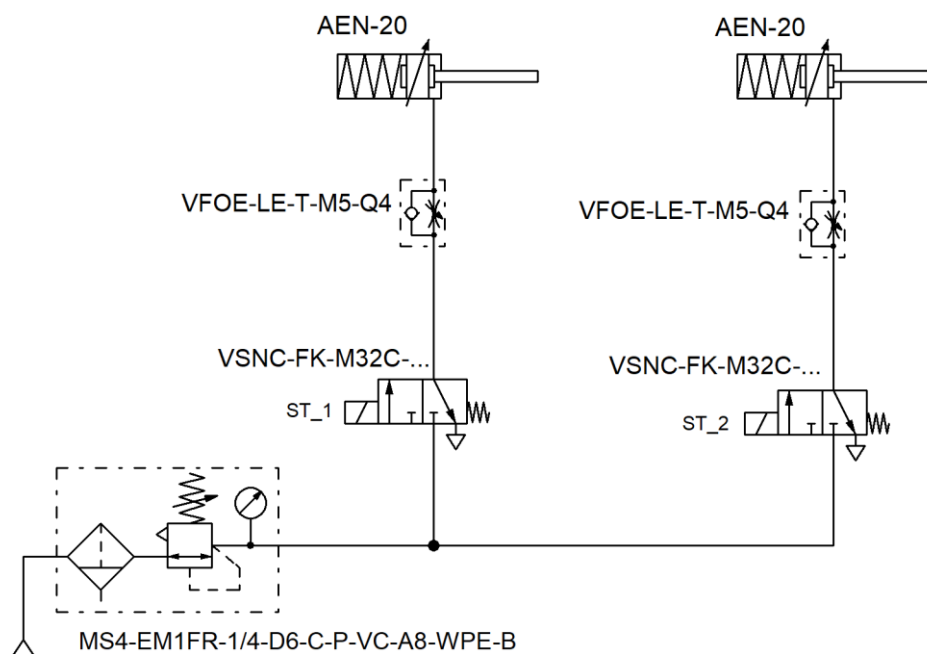
4.7.ábra: Tartó szerkezet modell

4.1.3.2. Ütközők

Annak biztosítására, hogy a robotkar által felvett munkadarabot ne nyomja oldalról emelés közben a sorban következő munkadarab, ütközőkre van szükség. Két darab monostabil pneumatikus munkahenger kerül elhelyezésre a szállítószalag mellett, amelyek zsilip szerű működéssel irányítják a munkadarabok mozgását. Erre a feladatra a Festo AEN-20 kompakt munkahengereket választottam. Az ütközők vezérlésére két darab 3/2-es, elektromos működtetésű útváltó szelepet alkalmaztam, amelyeknél a Festo VSNC-FK-M32C-RD-G18-1C1-S szelepeire esett a választásom.

Amikor a robotkar elvette a munkadarabot, akkor a két ütköző közötti munkadarab úgy kerül a megfelelő pozícióba, hogy az a szalag végéhez közelebbi ütköző visszahúzódik, majd amikor a szenzor jelez hogy elérte a véghelyzetet akkor újra működésbe lép. Ezek után a másik ütköző engedni a következő munkadarabot előre, majd egy időzítő lejártával újra kinyúl, így biztosítva hogy egyetlen munkadarab legyen a két ütköző közötti részben. Az időzítő a munkadarabok előrehaladási sebessége alapján van beállítva, tapasztalati érték. Ennek a folyamatnak a vezérlését a PLC egyik alprogramja végzi. Az ütközők pneumatikus kapcsolása megegyezik, amelyet az alábbi ábra mutat.

Designation	Description
VSNC-FK-M32C-...	3/n Way Valve
VSNC-FK-M32C-...	3/n Way Valve
VFOE-LE-T-M5-Q4	One-way flow control valve
VFOE-LE-T-M5-Q4	One-way flow control valve
MS4-EM1FR-1/4-D6-C-P-VC-A8-WPE-B	Air service unit
AEN-20	Cylinder, Single-acting, Return Spring in Piston Chamber
AEN-20	Cylinder, Single-acting, Return Spring in Piston Chamber
	Compressed air supply



4.9.ábra: Ütközők pneumatikus kapcsolása

4.1.3.3. Levegő fogyasztás

A munkahengerek esetében a kapcsolásonként elhasznált sűrített levegő:

$$V = \frac{d^2 * \pi}{4} * L = \frac{20^2 * \pi}{4} * 25 = 7,85 \text{ cm}^3$$

Ciklusonként 2 kapcsolás történik, tehát $109,9 \text{ cm}^3$ normál állapotú sűrített levegő kerül felhasználásra, 6 bar túlnyomást feltételezve. Percenként körülbelül 10 kapcsolás történik üzem közben, tehát 1099 cm^3 , vagyis $1,1 \text{ l/min}$ levegő áramlás szükséges. A vákuum megfogó esetén a választott vákuum generátor működés közbeni légfogyasztása 50 l/min . Működés

közben 5 másodpercig van szükség vákuumra ciklusonként, tehát átlagosan óránként 1250 liter sűrített levegőt fogyaszt.

Ezek az adatok alapján a Festo MS4-EM1FR-1/4-D6-C-P-VC-A8-WPE-B típusú levegő-előkészítő egységét választottam, amelynek 1500 l/min átfolyása bőven elegendő tesz a követelményeknek.

4.1.1. Szenzorok

A rendszer két helyen igényel szenzoros ellenőrzést, a paletta és a munkadarab elhelyezkedésének ellenőrzésére. Mindkét esetben lehetőség van a szenzorokat a mért tárgyhoz közel elhelyezni, így nincs szükség nagy hatótávolságra. Ebből következőn, valamint hogy a paletta és munkadarab is nem fémes anyagból készült, kapacitív szenzorokat választottam a feladathoz.

A választásom a Balluff BCS00PU szenzorra esett, amely PNP záró (NO) interfészű, 1-4 mm között képes érzékelni és 2 méter hosszú kábellel rendelkezik, elegendő téve a követelményeknek.



4.8.ábra: Kapacitív szenzor [19]

A munkadarab jelenlét ellenőrző szenzor a szállítószalag végén kerül elhelyezésre, amely így akkor jelez, ha a munkadarab elérte a szalag végén található ütközőt, ez által biztosítva a munkapozícióban való elhelyezkedést. A paletta ellenőrző szenzor közvetlenül a paletta pozícióját biztosító kialakítás mellett helyezkedik el. Ez a szenzor felel azért, hogy a rendszer ellenőrizni tudja, hogy a kezelő megfelelő módon helyezte el az üres palettát a csere után. A szenzorok üzemi feszültsége 24 V DC. Mindkét szenzor az I/O modulon keresztül ad jelet a PLC-nek.

4.2. A PLC programozása

A PLC a palettázó rendszer központja, ide vannak bekötve a folyamatot követő szenzorok, innen olvassa ki a távfelügyeleti weboldal az adatokat, valamint innen kapja a robotkar a vezérlő jeleket. Lényegében itt történik a rendszer jelenlegi állapotának nyomon követése és a döntés hozatal.

4.2.1. Feladatmegoldás

A program elkészítése előtt az első feladatomban volt a kommunikáció biztosítása a PLC és a robot vezérlő, az irányító felület, valamint a távfelügyeleti weboldal között. Nulladik lépésnek tekinthető a számítógép és a PLC közötti kapcsolat létrehozása a programok feltöltésére, valamint a virtuális kezelő felület csatlakoztatására. Ezt a Festo cég által készített FST nevű program telepítésével oldottam meg, amelyet a gyártó a saját FEC-CPX típusú PLC-k moduljaihoz készített. A kezelő felületemet a FEC Soft Control program adta, amely egy virtuális nyomógombokból és lámpákból álló program. Ezt a programot a 4.2.2. fejezetben részletezem.

A számítógép és PLC fizikai összekötésére egy UTP-UTP típusú kábelt alkalmaztam, amelyen keresztül egy TCP/IP protokollon alapuló kommunikációs csatornát hoztam létre. A számítógépet és a vezérlő felületet ugyanarra az IP címre kellett állítani, amelyet egy részből a számítógép hálózati beállításánál, más részből az FST-n belüli beállításoknál tudtam megtenni. A kapcsolat ellenőrzésére a szintén a Festo által készített Festo Field Device programot alkalmaztam.

Ezek után szükséges volt az alkalmazott eszközök jeleinek azonosítására, valamint hogy melyik csatornán csatlakoznak az I/O terminálhoz a PLC-n. Az FST programot online módba kapcsolva tudtam ezeket a jeleket tesztelni, amelyeket felvettem a PLC változó listájába. Ezeket az alábbi táblázatban részletezem.

Kimeneti jelek		
Kapcsolat	Szimbólum	Magyarázat
Robot	R_START	Robot ciklus start
	R_STOP	Ciklus vége stop
	P_OK	Paletta csere nyugtázása a robot felé
	POZ_J	Munkadarab jelenlét ellenőrzés
	R_V_STOP	Robot vészmegállító
Vezérlő	START_L	Start lámpa
	STOP_L	Stop lámpa
	P_FULL_L	Paletta tele lámpa
	EMSTOP_L	Vészmegállító lámpa
Weblap	R_COMM_F	Robot kommunikáció flag
	P_RUNN_F	Paletta ciklus futás flag
	P_FULL_F	Paletta tele flag
	C0	Ciklus munkadarab számláló
	C1	Összes munkadarab számláló
	W_D_F	Watcdog flag
Ütközők	ST_1	1-es munkahenger
	ST_2	2-es munkahenger

4.1.táblázat: PLC kimeneti jelek

Bemenő jelek		
Kapcsolat	Szimbólum	Magyarázat
Robot	R_COMM	Robot kommunikáció
	P_RUN	Palettázás ciklus futása
	P_FULL	Paletta tele
	P_SZ	Számláló jel
Vezérlő	START	Palettázás indítása
	STOP	Ciklus vége stop
	P_EMPTY	Paletta csere nyugtázása a PLC felé
	NYUGTA	Vésmegállítás után nyugtázás
	EMSTOP	Vésmegállítás
Szenzor	POZ_SZ	Munkadarab pozíció szenzor
	PAL_SZ	Paletta jelenlét szenzor
	VAC_SZ	Vákuum szenzor

4.2.táblázat: PLC bementei jelek

A PLC tervezett feladatának ellátására több program létrehozására volt szükségem, amelyek felsorolva:

- Alap program
- Lámpa villogás program
- Hiba program
- Szállítópálya program

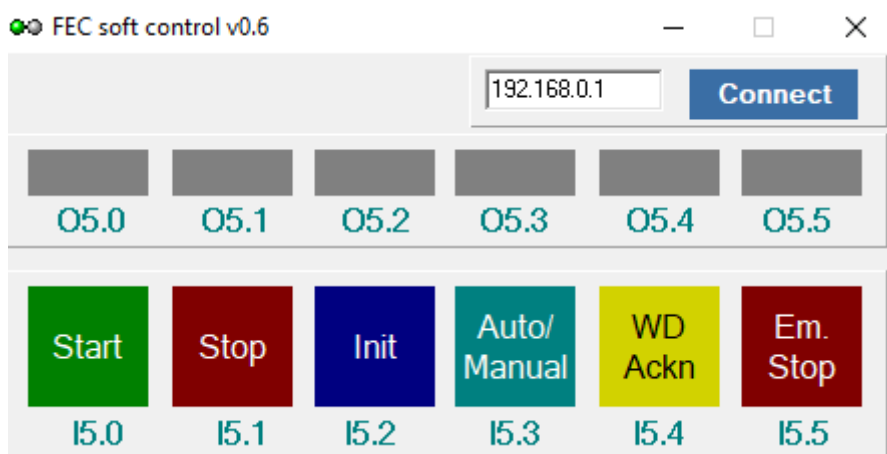
Ezek közül a legfontosabb az alap program, amely felelős a beérkező jelek kezeléséért. Ezt a programot a jelentősége miatt a 4.2.3. fejezetben fejtem ki bővebben.

A lámpa villogás program egész egyszerűen a megtelt palettára való figyelem felhívásáért felelős, a virtuális kezelő felület egy lámpájának működtetésével.

A hiba program az alap programban elhelyezett watchdog időzítő lejárta esetén szükséges, amely a beérkező jelek alapján azonosítja a rendszer hibáját, amely információt aztán hibakód formájában továbbít a kezelő felé.

A szállítópálya program a következő munkadarab a felvételi pozícióba való eljuttatásáért felelős, amelyhez a szállítópálya mozgását és az ütközőt vezérli.

4.2.2. Kontroll panel



4.10.ábra: FEC Soft Control

A képen ábrázolt virtuális vezérlő panel a FEC Soft Control program hozta létre, amely a korábbi fejezetekben leírtak alapján a PLC-hez van csatlakoztatva. A palettázó folyamathoz 5 gombot és 4 lámpát használtam fel, amelyek lehetővé teszik a kezelő számára a beavatkozást a folyamatba.

Az alábbi táblázatokban felsoroltam az felhasznált gombokat és lámpákat, valamint a PLC kommunikációhoz alkalmazott ki- és bementi jeleket.

Virtuális gombok		
Jel	Megnevezés	Funkció
I5.0	START	Folyamat indítás
I5.1	STOP	Ciklus vége stop
I5.2	P_EMPTY	Paletta nyugtázás csere után
I5.4	NYUGTA	Nyugtázás vészmegállás után
I5.5	EMSTOP	Vészmegállás

4.3.táblázat: Vezérlő gombok

Virtuális lámpák		
Jel	Megnevezés	Funkció
O5.0	START_L	Start lámpa, rendszer készenlélet jelez az indításnál
O5.1	STOP_L	Ciklus vége stop érvényességét jelzi
O5.2	P_FULL_L	Villogással jelzi a paletta telítettségét
O5.5	EMSTOP_L	A vészmegállás érvényességét jelzi

4.4.táblázat: Jelző lámpák

A rendszer indításakor a PLC ellenőrzi a bemenő jelek alapján az üzemképességet, ha mindent jónak talál akkor felkapcsolja a Start lámpát (START_L), ezzel jelezve a készültséget a kezelő felé. A START gomb megnyomásával elindul a munka ciklus, amely egészen a paletta megteléséig fut, amelyet a paletta telítettség lámpa (P_FULL_L) villogása jelez a kezelő felé. A paletta cseréje után a villogó lámpa alatti P_EMPTY gomb megnyomásával lehet jelezni a rendszernek, amely utána folytatja a munkaciklust.

A folyamat megállítására két lehetőséget hoztam létre. Az első a STOP gomb, amely a jelenleg futó ciklus befejeződése után alaphelyzetben megállítja a folyamatot. A gomb megnyomása után a stop lámpa (STOP_L) világítása jelez hogy a rendszer észlelte a parancsot és a megfelelő pozícióra vár. A másik megállítási lehetőség az EMSTOP gomb, amely azonnal megállít minden mozgást, a robot és a PLC programban is. Ennek a vészmegállításnak az érvényességét az vészstop lámpa (EMSTOP_L) jelzi a kezelő felé. Mivel a stop gombok reteszelvek, ezért a

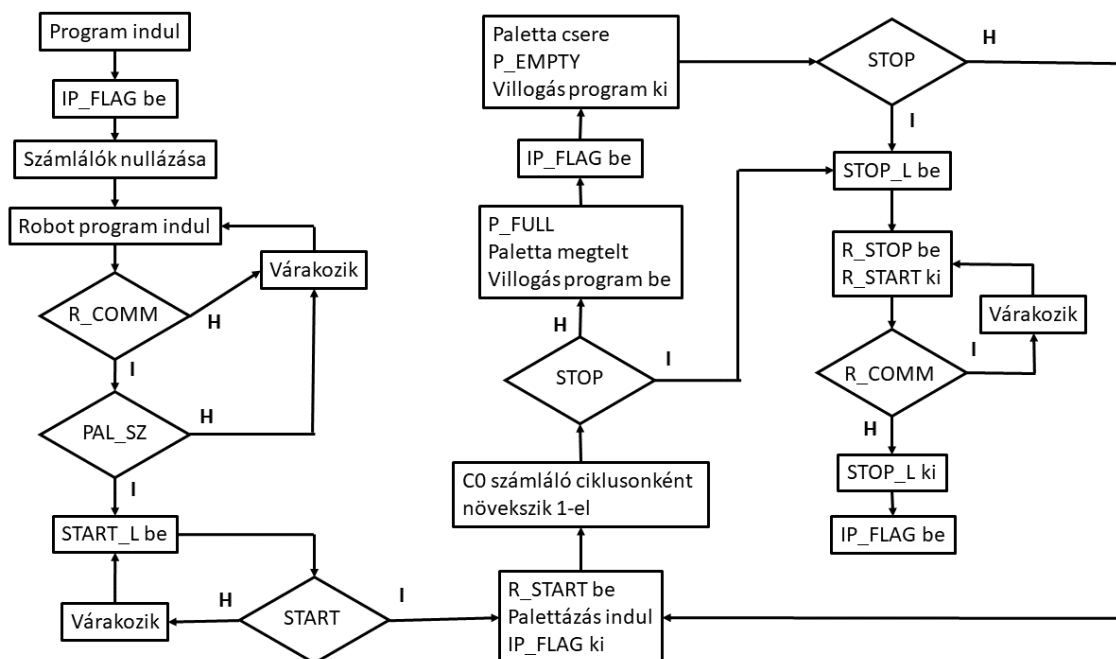
vészmegállítás feloldásához először újra meg kell nyomni az EMSTOP gombot, majd a NYUGTA gombbal kell jelezni a rendszernek, hogy folytatható a munka.

4.2.3. PLC alap program

A PLC alap programja felel a munkaciklus irányításáért és ellenőrzéséért. Ennek érdekében a programban IF-THEN parancsokat hoztam létre, amelyekben a megfogalmazott feltétel rendszereket folyamatosan vizsgálja a program. A program egyszerűbb átláthatósága érdekében több részre tagoltam, ezek felsorolva:

- Indítási protokoll
- Alaphelyzet ellenőrzés
- Ciklus indítás
- Ciklus vége stop
- Paletta számláló
- Paletta csere
- Vészmegállítás
- Flag-ek működtetése
- Watchdog időzítő

Ezen részek mindegyike egy-egy funkció ellátásáért felel, amelyeket az IF utasításban megfogalmazott feltételek teljesülése esetén hajt végre a program. A program egy indítási protokoll flag-et használ a két fő működési mód megkülönböztetésére. Ezek a munkaciklus, valamint az indítás vagy paletta csere időszakai. Az indítási protokoll flag felkapcsolása esetén a robotkar alaphelyzetben áll, mozgásra csak a flag lekapcsolása után kerül sor, ezzel egyben megkülönböztetve a robot munkaterébe való benyúlás biztonságos időszakát.



4.11.ábra: PLC alap program hatásvázlat

Az indítási protokoll a PLC elindítása esetén felel az indítási protokoll flag (IP_FLAG) bekapcsolásáért és az alkalmazott számlálók nullázásáért.

Az alaphelyzet ellenőrzés felel a robot programmal való kommunikáció működésének, valamint a munkadarab és paletta jelenlétének ellenőrzéséért. Csak ezeknek a feltételeknek a teljesülése után van lehetőség a kezelő számára a munkaciklus elindítására, amelyet a start lámpa felkapcsolásával jelez a program.

A ciklus indítás a felgyulladt start lámpa esetén megnyomott START gomb esetén megy végbe, kiadva a robotnak az indulási engedélyt, valamint lekapcsolva a PLC programon belül az indítási protokoll flag-et.

A ciklus vége stop a STOP gomb megnyomása esetén lép érvénybe, ez egyszerre ad jelet a robotnak a jelenlegi ciklus végén való megállásra, valamint kifelé a kezelőnek a parancs érvényességére a stop lámpa alkalmazásával.

A paletta számláló nyomon követi a cikluson belül a jelenleg pakolt munkadarab számát. Ezt a robot programban elhelyezett, a PLC felé kimeneti jel figyelésével teszi. Minden esetben amikor ez a jel 1-re vált, a C0 számláló növekszik 1-gyel. Annak érdekében, hogy csak a felkapcsoláskor változzon a számláló értéke, egy edge flag-et alkalmaztam, amely a számláló jel felkapcsolása esetén szintén felkapcsol. Ezen felül a program a robottól kapott paletta telítettség jellel figyeli hogy mikor éri el a munkaciklus a paletta maximális értékét, ekkor a C0 számláló jelenlegi értékét hozzá adja a C1 számláló értékéhez, ez által nyomon követve az elindulás óta átpakolt munkadarabok számát. Ezen felül a paletta telítettsége esetén elindítja a paletta lámpa villogás programot.

A paletta csere rész az új paletta behelyezése után egy szenzoron keresztül ellenőrzi a paletta jelenlétét, majd a paletta nyugtázás gomb megnyomásakor jelet ad a robot felé, hogy a következő munkaciklus elindításhoz teljesültek a feltételek.

A vészmegállás az EMSTOP gomb megnyomása esetén lép érvénybe, amelynek hatására parancsot ad a robotnak a lehető legrövidebb úton való megállásra. Ezen felül a PLC-n belül is leállít minden folyamatot, ideértve a szállítópálya vezérlését is. A művelet újra indítása csak az EMSTOP reteszelésének oldásával és a NYUGTA gomb megnyomásával lehetséges.

A hibakereséshez létrehoztam egy Watchdog időzítőt, amely lejártakor indul el a hibaprogram. A palettázási cikluson belül mindig a számláló növekedése esetén indul újra az időzítő, amelyre 60 másodpercet állítottam be. A paletta csere során történő időzítő lejárat esetén, ha már jelzett a robot a paletta telítettségéről, akkor nem indul el a hibaprogram. A paletta csere utáni nyugtázás során indul újra a Watchdog. A program ezen részében történik a vákuum rendszer ellenőrzése, amely a watchdog időzítőtől függetlenül működik.

Az alap program utolsó része felelős a beérkező jelek alapján változtatni a PLC-ben alkalmazott flag-eket. Ezek utána a PLC a belső működésében használja fel a folyamat követésére, valamint a távfelügyeleti weboldal ezeken keresztül nyer információt a rendszer jelenlegi állapotáról.

4.3. A robot programozása

A robotkar a rendszer végrehajtó szerve, ez felel a munkadarabok eljuttatásáért a kívánt pozícióba a palettán. Ebben a fejezetben tárgyalom a robot program felépítését, valamint a palettázás során végzett feladatait.

4.3.1. A robot felépítése

A palettázási feladat ellátáshoz egy 5 tengelyes RRR típusú Mitsubishi RV-2AJ humanoid robotkart használtam fel, amelyet a tanszék szolgáltatott a szakdolgozatom elkészítéséhez. A robotkar 5 rotációs csuklóval rendelkezik és a maximális terhelhetősége 2 kg, a munkaterének maximális elérési távolsága 410 mm. A robotkart egy pneumatikus vákuum megfogóval szereltem fel a feladathoz. A robotkar további lényeges elemei a vezérlőszekrény, a teach pendant, az I/O kommunikációs egység, egy I/O kábel, valamint a pneumatikus interfész a megfogó működtetéséhez.



4.12. ábra: RV-2AJ robotkar [14]

4.3.2. Feladatmegoldás

Az első feladat a robot és PLC közötti kommunikáció biztosítása volt, amelyet a robotvezérlőhöz kapcsolt I/O terminál segítségével oldottam meg. A robot által alkalmazott ki- és bemeneti jelek az alábbi táblázatokban vannak feltüntetve.

Kimeneti jelek		
Robot program jelölés	PLC program jelölés	Magyarázat
M_OUT(10)	R_COMM	Robot kommunikáció
M_OUT(11)	P_RUN	Palettázás ciklus fut
M_OUT(12)	P_FULL	A paletta megtelt
M_OUT(13)	P_SZ	Számláló jel

4.5.táblázat: Robot kimenő jelek

Bemeneti jelek		
Robot program jelölés	PLC program jelölés	Magyarázat
M_IN(10)	R_START	A palettázás indítása
M_IN(11)	R_STOP	Ciklus vége stop
M_IN(12)	P_OK	Paletta nyugtázás csere után
M_IN(13)	POZ_J	Munkadarab jelenlét ellenőrzés
M_IN(14)	R_V_STOP	Robot vészstop

4.6.táblázat: Robot bemenő jelek

Az eszközök összekapcsolása után a következő feladatom volt definiálni a robotprogramban az alkalmazott vákuum megfogó esetén a TCP-t (Tool Center Point). Ez azért volt fontos, hogy a munkarab felvételekor biztosítsam az optimális működést a vákuum korong számára.

A következő lépés a paletta méretének és pozíciójának definiálása volt, amelyet a robot dokumentációjában leírt struktúra szerint tettem a PLT utasítással. Ennek részeként felvettem a paletta sarokpontjait, majd a sorok számának és az egy sorban elhelyezkedő munkadarabok mennyiségének megadásával a robot program kiszámította az elhelyezési pontokat a palettán.

Ezek után következett a robot a munkaciklus alatt megtett útjának felvétele, amelyet 3 pont rögzítésével tettem, amelyek a következők:

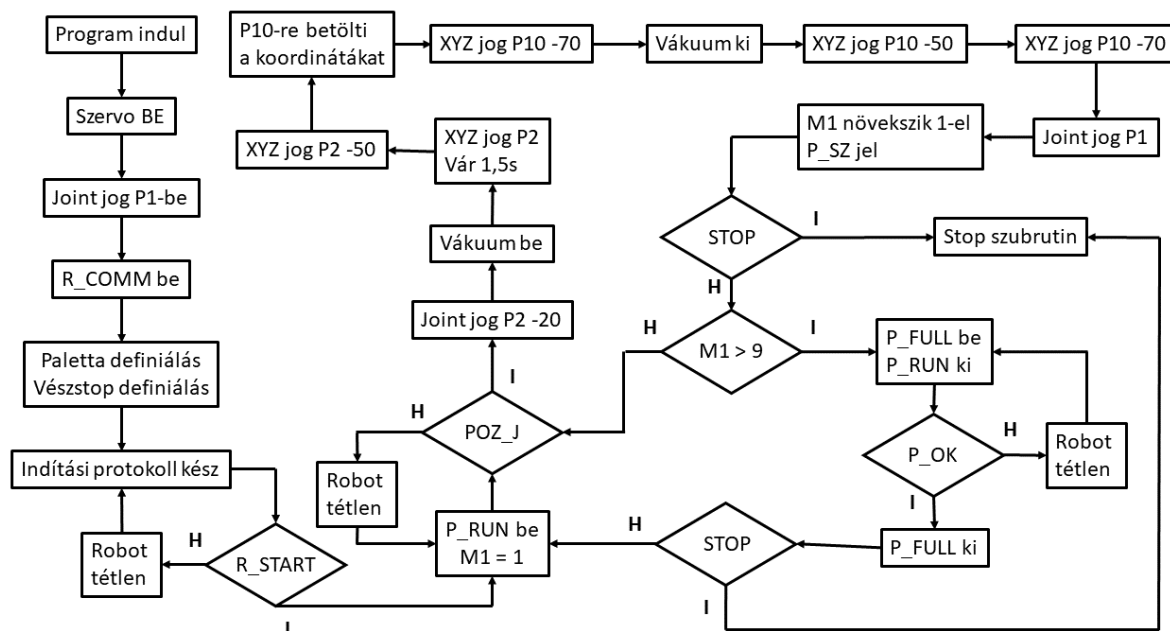
- P1, a robot alaphelyzet pontja, ahol a ciklus kezdődik és végződik
- P2, a felvételi pont, ahonnan a munkadarabot kiemeli
- P10, ahova a munkadarab kerül a palettán, ez a pont a ciklus alatt változik

A robot program egyszerűsítésének érdekében minden alkalommal amikor a robotkar egy új munkadarabot fog meg a P10-es pont megváltozik, az adott munkadarab cikluson belüli számának függvényében.

4.3.3. Robotprogram

A robot feladatának elvégzésére egyetlen programot hoztam létre. A program első szakasza felel a robot alaphelyzetbe állításáról, ami azt jelenti, hogy bekapcsolja a szervót, a robot a P1 pontba mozog és kikapcsolja a vákuumot, valamint egy jelzéssel (R_COMM) ellenőrzi a PLC felé a kommunikáció működését. A programba nem került külön alaphelyzetre állítás, mivel minden újra indításkor ez megtörténik a program elején, valamint nincs olyan megállási vagy vészmegállási helyzet, ahol ez nem igaz.

A következő két lépés a program kritikus része, itt kerül definiálásra a paletta és a vészleállítás. A paletta definiálása a sarokpontok felvételével, valamint a sorok számának és hosszának megadásával történik, amely alapján a program kiszámítja az elhelyezési pontokat. A jelen esetben egy 3x3-as, egy rétegű paletta került felvételre, amelyre úgynevezett „zigzag” útvonalú sorrendben kerülnek elhelyezésre a munkadarabok.



4.13.ábra: Robot program hatásvázlat

A vészleállításhoz egy ACT került definiálására, amely a PLC-től kapott vészleállító jel (M_IN(14)) esetén megállítja a robotkar mozgását és program végén található Stop szubrutinra ugrik, ahol a szervó kikapcsolásra kerül. Ez azt jelenti, hogy a vészleállítás után újra indítás során a program első lépései viszik a kezdő pozícióba a robotkart. Az ACT utasításban továbbá beállítottam az S paraméterrel, hogy a lehető legrövidebb úton álljon meg a robotkar.

Ezek után a robotprogram fő része következik. A robot először is vár addig, amíg megkapja a PLC-től a jelet (R_START) az indulásra, amelyet követően kiküld egy jelet (P_RUN) a palettázó ciklus futásának igazolására.

A munkaciklust egy FOR ciklussal oldottam meg, amely számon tartja, hogy hányadik munkadarabot helyezi éppen a palettára. Ezen felül létre hoztam egy számlálót kimeneti jelet, amely továbbítja ezt az információt a PLC felé, onnan pedig a távfelügyeletre. Amikor ez a számláló eléri a beállított értéket, a robot jelez (P_FULL) a kezelő felé a paletta telítettségéről, majd a kezdeti pozícióban várakozik addig, amíg jelet (P_OK) nem kap az új paletta behelyezéséről. A cikluson belül létre hoztam egy P10 pontot, amelynek a helyzetét mindig a jelenlegi munkadarab sorszáma alapján változtatok. Ez úgy lehetséges, hogy a korábban

definiált palettán a program a megadott adatok alapján már létrehozott pontokat, amelyhez sorszámot rendelt a palettán való elhelyezkedésüktől függően. Minden alkalommal ebbe a P10-es pontba töltöm fel a koordinátákat a palettára való elhelyezéshez.

Minden ciklus során a P1 alaphelyzeti pontból indul a robotkar, ahonnan a P2 pont fölé áll, megtartva egy biztonsági távolságot. Ez után bekapcsol a vákuum, majd a robotkar csökkentett sebességgel a P2 pontba mozog. A P2 pont elérése után vár 1,5 másodpercet a biztos megfogás érdekében. Innen újra felemelkedik az ütközés elkerülésére, majd a munkadarab sorszáma alapján megváltozott P10 pont fölé mozog, amely után megint csökkentett sebességgel lehelyezi a munkadarabot a palettára, amely a programban a P10 fölött van definiálva, mert figyelembe kellett vennem a munkadarab magasságát is. Ez a távolság a paletta felszínén definiált P10 és a megfogó TCP-je közötti távolság, megegyezik a munkadarab magasságával. A lerakott munkadarab orientációját a program a paletta definiálásánál felvett sarokpontok orientációjából számítja ki.

Bár a rendszer képes lehetne a teljesen automata működésre abban az esetben, ha a paletta szintén egy szállító szalagon lenne elhelyezve, a jelenlegi módszer amit alkalmaztam egy kezelőre támaszkodok a paletták cseréjéhez, amelynek elvégzését egy gombnyomással tud nyugtázni.

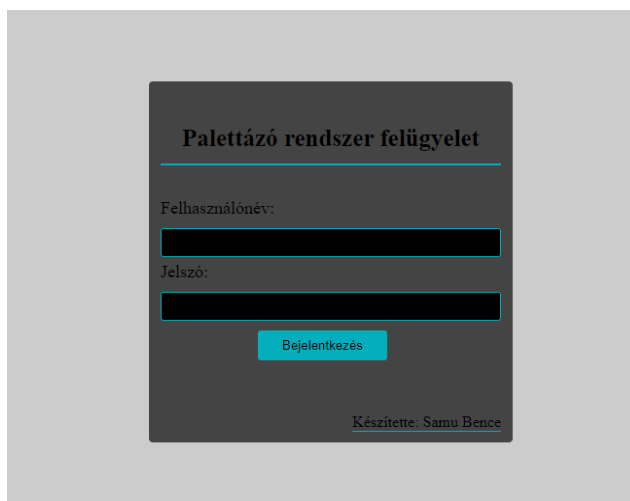
Természetesen szükség van a megállás beprogramozására is, amelyet egy ciklus vége stoppal biztosítottam. A program futása során két helyen kerül ellenőrzésre, hogy megnyomták-e a stop gombot, a munkaciklus végén, valamint a paletta csere után új munkaciklus kezdeténél. A ciklus végén található ellenőrzésre a palettázási folyamat alatt kiadott utasítás esetén van szükség, amellyel így nem kell a paletta megteléséig várni.

Felmerülhet olyan eset, hogy amíg a robot a paletta cseréjére várakozik, akkor döntünk a leállításról. Ebben az esetben az új paletta behelyezésének nyugtázása után kerül sor a programban az ellenőrzésre, elkerülve egy fölösleges ciklus elvégzését. Mindkét esetben a program végére található lépésekre ugrik, ahol kikapcsolja a szervót és leállítja a programot.

4.4. Távfelügyelet

A távfelügyelet megvalósításához egy HTML weboldalt készítettem, amely információt szolgáltat a rendszer jelenlegi állapotáról néhány megfigyelt változó segítségével. Ennek az elérésére szükségem volt egy java script fájlra, amely képes a PLC-ből a szükséges információk kiolvasására és átadására a weboldal számára. Ezt a tanszékről kapott fst.js fájl segítségével tettem, amellyel a HTML fájlban meg tudtam hivatkozni a számomra szükséges Flag változókat és számlálókat.

Annak érdekében, hogy a távfelügyeletet csak jogosult személyek érhesék el, egy bejelentkező oldalt hoztam létre.



4.14.ábra: Távfelügyelet bejelentkezés

A bejelentkezésnél egy Admin és egy User felhasználót definiáltam, amelyek eltérő mennyiségű információt látnak a weboldalon. Az Admin felhasználó mindent lát, azonban a User felhasználó csak a palettázás futását, a számlálót, valamint a hibát és a hibakódokat láthatja.

A weboldal létrehozásakor az első lépésem volt a rendszeres frissítés megoldása, amelyet a jelenlegi dátum kiírásával, valamint egy 3 másodpercenként történő újra töltés beállításával tettem. Ez biztosította, hogy mindig az aktuális adatok jelenjenek meg a weblapon.

A HTML kódban a PLC-ből a megfelelő megnevezések használatával lehet információt kinyerni. Flag-ek esetén ennek a struktúrája: <FSTCI dmw1>, ahol a szám a PLC-ben alkalmazott Flag sorszáma utal. A számlálók esetén dmw1 helyett dzw1 lesz a megnevezés.

A palettázás jelenlegi állapota	
Megnevezés	Állapot
Robot kommunikáció	
Palettázás ciklus	
Palettán elhelyezett darabok száma	
Paletta telítettsége	
Összes álpakolt darab	
Watchdog	

4.15.ábra: Weboldal táblázat

A weboldal fő részét egy táblázatos formában alkottam meg amelynek a celláin belül helyeztem el a megfigyelt változók megnevezését és jelenlegi értékét. Összesen 6 adatot lehet a weboldalon keresztül nyomon követni, ezek közül négyet Flag változókon keresztül kommunikál a PLC az oldal felé, ahol *switch* utasításokkal alakul át az információ szöveggé. Ez a négy változó a robot kommunikáció, paletta ciklus futás, paletta telítettség és a watchdog. A weboldalon található még 2 számláló is, amelyek a jelenlegi palettán elhelyezett munkadarabokat, valamint az indítás óta átrakodott munkadarabok számát követik nyomon. Mivel ez az információ egy szám, ezért ebben az esetben nincs szükség a *switch* utasításra, elegendő a számlálók meghívása a megfelelő megnevezéssel.

A távfelügyeleti rendszer nem rendelkezik beavatkozási funkcióval a palettázási folyamatba, csak a rendszer jelenlegi állapotáról ad információt. A távfelügyeleti weboldal HTML kódja a mellékletekben található.

5. Továbblépési lehetőségek

Az elkészült palettázó rendszerben több lehetőség is van a további fejlesztésre a folyamat hatékonyságának növelésére.

A jelen megoldás manuális beavatkozást igényel a paletta megtelése esetén, amit azonban egy második szállítópálya beépítésével automatizálni lehetne. Ebben az esetben egy további szenzor beépítésére lenne szükség, amely a megérkező paletta felszínét vizsgálná, ellenőrizve hogy üres és készen áll a rakodásra. Ezen felül módosításra van szükség a paletta helyének ellenőrzésénél, mivel nincs manuális ellenőrzés a paletta helyes behelyezésére.

További előre lépés lehet a megépített rendszer cellájának biztonsági rendszerének megépítése. A jelenlegi kialakításban nincs ellenőrzés a robot munkaterében tartózkodó személyek jelenlétére, amely megoldható lenne a robot munkaterének elkülönítésével, valamint a szükséges nyílások fényfüggönnyel való ellátásával. Ez a biztonsági rendszer szükségessé tenné a robot megállítására szolgáló parancsok áttervezését.

6. Gazdasági számítás

A feladat elvégzéséhez a tanszék által biztosított robotkart és PLC-t használtam fel, amelyeket így nem vettem figyelembe a gazdasági számításnál. A programozáshoz bármilyen számítógép alkalmazható, amely rendelkezik UTP csatlakozóval és Windows operációs rendszert futtat, az alkalmazott szoftverek mindegyike ingyenesen elérhető, ezért ezek sem kerültek beszámításra. Azonban a feladatmegoldás során kiválasztottam egy szállítópályát, két szenzort, egy vákuumgenerátort és négy vákuumkorongot, amelyek mindegyikét belevettem az alábbi számításba. A két darabból álló vákuum tartó szerkezet legyártása szintén bekerült a számításba. A pneumatikus rendszerhez tartozó levegő-előkészítő egységet és fojtó visszacsapó szelepet a tanszék szolgáltatta, ezeket sem adtam hozzá a számításhoz. A feladaton 6 héten keresztül heti egy nap 8 órán keresztül dolgoztam

Megnevezés	Darabszám	Egységár	Összesen
Munkaórák	48 óra	11 000 Ft	528 000 Ft
BCS00PU szenzor	2 db	28 992 Ft	57 984 Ft
Vákuumkorong	4 db	7 290 Ft	29 160 Ft
Vákuum generátor	1 db	365 990 Ft	365 990 Ft
Szállítópálya	1 db	5 000 000 Ft	5 000 000 Ft
Vákuum tartó	2 db	20 000 Ft	40 000Ft
			6 021 134 Ft

7. Összegzés

A szakdolgozatom elkészítése során megterveztem egy, az ipari környezetben felmerülő rakodási feladat ellátására egy palettázó rendszert. A munkám során megterveztem a folyamat menetét, amelynek kivitelezésére elkészítettem a robotprogramot. A munkadarabok méretei alapján megterveztem a palettát, amelynek jelenlétének ellenőrzésére egy kapacitív szenzort építettem be. A rendszer követelményeinek megfelelően kiválasztottam a szükséges kiegészítő elemeket. A munkadarabok megfogására megterveztem egy vákuum megfogó rendszert. Kiszámítottam a szükséges vákuum korong átmérőt, amely alapján kiválasztottam a megfelelő korongokat, valamint a vákuum generátort is. A vákuum megfogó a robotkarhoz való rögzítésére terveztem egy két elemből álló tartó szerkezetet, amely egyben biztosítja pneumatikus kapcsolást is. A munkadarabok elszállítását a robotkar munkaterébe egy görgős szállítószalaggal oldottam meg, amely hajtás nélküli kivitelben, a munkadarabok lendületét és a gravitációt használja a szállításra. A munkadarabok pozíciójának ellenőrzésére kiválasztottam egy kapacitív szenzort, a mozgásuk irányítására pedig két pneumatikus munkahengert alkalmaztam. Elvégeztem a pneumatikus alkatrészek levegő fogyasztás számítását. A palettázó rendszer vezérlésére egy PLC-t használtam fel, amelyet egy I/O modulon keresztül kötöttem össze a szenzorokkal, aktuátorokkal és a robot vezérlésével. Elkészítettem a PLC programokat a beérkező jelek kezelésére, a kezelő irányába való kommunikációra és a hibakeresésre. A palettázó rendszer pillanatnyi állapotának leolvasására létrehoztam a távfelügyeleti weboldalt. Az információk bizalmasságát egy bejelentkező oldal létrehozásával védtem, amely a felhasználói profiltól függően különböző mennyiségű információkat tesz elérhetővé. Elkészítettem a rendszer elektromos és pneumatikus kapcsolási rajzait. Elkészítettem a felhasznált alkatrészek darabjegyzékét, valamint a gazdasági számítást.

8. Summary

During the preparation of my thesis, I designed a palletizing system for handling a material handling task in an industrial environment. Throughout my work, I planned the process and created the robot program for its execution. Based on the dimensions of the workpieces, I designed the pallet and integrated a capacitive sensor to verify its presence. I selected the necessary auxiliary components in accordance with the system's requirements. For gripping the workpieces, I designed a vacuum gripping system. I calculated the required vacuum cup diameter and selected the appropriate cups and vacuum generator accordingly. To attach the vacuum gripper to the robot arm, I designed a two-part mounting structure that also provided pneumatic connection. For transporting the workpieces to the robot's workspace, I implemented a roller conveyor system in a non-driven design, utilizing the momentum of the workpieces and gravity for transportation. I chose a capacitive sensor to control the position of the workpieces and used two pneumatic actuators to manipulate their movement. I performed calculations for the air consumption of the pneumatic components. To control the palletizing system, I used a PLC (Programmable Logic Controller) connected to sensors, actuators, and the robot through an I/O module. I created PLC programs to handle incoming signals, communicate with the operator, and for troubleshooting. To monitor the current status of the palletizing system, I developed a remote monitoring web page. I protected the confidentiality of the information by implementing a login page that provided different levels of information access based on the user profile. I prepared electrical and pneumatic schematics for the system and created a bill of materials for the components used. Additionally, I conducted an economic analysis.

9. Felhasznált irodalom

- [1] Kulcsár Béla (2012): *Robottechnika I.* Budapest: Typotex Kiadó.
- [2] Balpataki Antal, Bécsi Tamás, Károly József, Márton Gergely, Szentannai Gábor (2012): *Járműhidraulika és -pneumatika.* Budapest: Typotex Kiadó.
- [3] Ferenczi Ildikó (2018): *Automatizálási alapismeretek 1.rész: Vezérléstechnika.* Nyíregyháza: Nyíregyházi Egyetem.
- [4] Bendekovits Z., Kóbor J., Pintér J. (2006): *Gépipari Automatizálás.* Győr: Széchenyi István Egyetem.
- [5] ISO 8373:2021 Robotics Vocabulary
- [6] Dr. Szabó Zsolt, Budai Csaba, Dr. Kovács László, Dr. Lipovszki György (2014): *Robotmechanizmusok.* Budapest: BME MOGI.
- [7] Forgó Zoltán (2009): *Műszaki Tudományos Füzetek. Bevezetés a Mechatronikába.* Kolozsvár: Erdélyi Múzeum Egyesület.
- [8] Veres György (2012): *Hidraulika és pneumatika.* Veszprém: Pannon Egyetem.
- [9] Laczik Bálint (2012): *Robottechnika.* Tatabánya: Edutus Egyetem.
- [10] Huba A., Aradi P., Czmerk A., Lakatos B., Chován T., Varga T. (2014): *Mechatronikai berendezések tervezése.* Budapest: BME-MOGI.
- [11] Ágoston A., Kohut M., Kurutz K. (1994): *Irányítástechnika II.* Budapest: Műegyetemi Kiadó.
- [12] Németh, Andrew. (1989): Robotizált anyagmozgatás a szerelésben (Szerelőcella). *Gépgyártástechnológia*, 29(5), 229-235.
- [13] AI-Robotics (2023): Ha a robotok végzik a munkát. *Műszaki Magazin*, 2023(10), 46.

[14] 4.12.ábra forrás:

https://www.eurobots.net/media/robot/det_259_robot_mitsubishi_melfa_rv-2aj.jpg

[15] 3.4.ábra forrás: <https://vmek.oszk.hu/07300/07377/07377.pdf> 139. oldal

[16] 3.5.ábra forrás: <https://barth-elektronik.com/out/pictures/master/product/1/9640-0544-A.JPG>

[17] 3.7.ábra forrás: https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcT60mRQ5fs9hA81tvjbUZ_JSvrofseTd7WvLw&usqp=C
[AU](#)

[18] 4.5.ábra forrás:

https://www.festo.com/media/pim/346/D15000100137346_488x366.jpg

[19] 4.8.ábra forrás: <https://www.balluff.com/hu-hu/products/BCS00PU>

[20] 3.1.ábra forrás: https://assets.robots.com/general/_800x418_crop_center-center_82_none/3_Industrial_Robots.png?mtime=1510709448

[21] 3.2.ábra forrás: <https://scioly.org/wiki/images/8/8c/Robots.jpg>

[22] 3.5.ábra forrás: <https://www.xpneumatic.com/wp-content/uploads/2017/07/3HP-200L.jpg>

NYILATKOZAT

a szakdolgozat nyilvános hozzáféréséről és eredetiségéről

A hallgató neve: Samu Bence
A Hallgató Neptun kódja: AO7LOK
A dolgozat címe: Palettázó ipari robot programozása és távfelügyeleti rendszerének tervezése
A megjelenés éve: 2023
A konzulens intézetének neve: Műszaki Intézet
A konzulens tanszékének a neve: Mechatronika tanszék

Kijelentem, hogy az általam benyújtott szakdolgozat egyéni, eredeti jellegű, saját szellemi alkotásom. Azon részeket, melyeket más szerzők munkájából vettem át, egyértelműen megjelöltem, és az irodalomjegyzékben szerepeltettem.

Ha a fenti nyilatkozattal valótlan 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: Hatvan, 2023 év november hó 3 nap

Samu Bence

Hallgató aláírása

KONZULTÁCIÓS NYILATKOZAT

A SAMU BENCE (név) (hallgató Neptun azonosítója: A07FL0K) konzulenseként nyilatkozom arról, hogy a szakdolgozatot¹ áttekintettem, a hallgatót az irodalmi források korrekt kezelésének követelményeiről, jogi és etikai szabályairól tájékoztattam.

A záródolgozatot/szakdolgozatot/diplomadolgozatot/portfóliót a záróvizsgán történő védelemre javaslom / nem javaslom².

A dolgozat állam- vagy szolgálati titkot tartalmaz: igen nem^{*3}

Kelt: Székesfehérvár, 2023. év november hó 2. nap


Belső konzulens

¹ A megfelelő dolgozattípus meghagyása mellett a többi típus törlendő.

² A megfelelő aláhúzendő.

³ A megfelelő aláhúzendő.

1. Melléklet

```
10 SERVO ON
20 M_OUT(10)=1
30 MOV P1
40 HOPEN 1
50 DEF PLT 1, P3, P4, P5, , 3, 3, 1
60 DEF ACT 1, M_IN(14)=1 GOSUB 400,
70 WAIT M_IN(10)=1
80 M_OUT(11)=1
90 M1=1
100 IF M_IN(11)=1 THEN GOTO *STOP
110 M_OUT(12)=0
120 FOR M1=1 to 9
130 M_OUT(13)=0
140 WAIT M_IN(13)=1
150 MOV P2, -20
160 OVRD 10
170 HCLOSE 1
180 MVS P2
190 DLY 1.5
200 MVS P2, -50
210 OVRD 70
220 P10=(PLT1, M1)
230 MOV P10, -70
240 OVRD 10
250 MVS P10, -50
260 HOPEN 1
```

```
270 DLY 0.5
280 MVS P10, -70
290 OVRD 70
300 MOV P1
310 M1=M1+1
320 M_OUT(13)=1
330 IF M_IN(11)=1 THEN GOTO *STOP
340 IF M1=9 THEN M_OUT(12)=1
350 NEXT
360 WAIT M_IN(12)=1
370 GOTO 90
380 *STOP
390 SERVO OFF
400 M_OUT(10)=0
410 M_OUT(11)=0
420 END
```

2. Melléklet

- Alap program

"" Indítási számláló nullázás

```
IF          NOP
THEN LOAD   V9
  TO        CP0      'Számláló végérték
SET         C0      'Cikluson belüli SZÁMLÁLÓ
SET         C1      'Összes darab számláló
SET         C2      'Hibakód
SET         IP_FLAG  'Indítási protokoll FLAG
SET         P3      'Szállítópálya PROGRAM
```

"" Elindítás alaphelyzet ellenőrzés

```
IF          R_COMM    'Robot kommunikáció JELZÉS
AND         IP_FLAG    'Indítási protokoll FLAG
AND         PAL_SZ     'Paletta jelenlét szeszor
THEN SET    START_L    'Start lámpa JEL
```

"" Ciklus indítás

```
IF          START     'Start GOMB
AND         START_L    'Start lámpa JEL
AND         IP_FLAG    'Indítási protokoll FLAG

THEN
  RESET     START_L    'Start lámpa JEL
  SET       R_START    'Robot start/ Robot üzem PARANCs
  RESET     IP_FLAG    'Indítási protokoll FLAG
```

"" Ciklus vége STOP

```
IF          N   STOP    'Stop GOMB
AND         R_COMM  'Robot kommunikáció JELZÉS
THEN SET    STOP_L    'Stop lámpa JEL
  RESET     R_START    'Robot start/ Robot üzem PARANCs
  SET       R_STOP     'Robot ciklus vége stop PARANCs
```

```
IF          STOP_L    'Stop lámpa JEL
AND         N   R_COMM  'Robot kommunikáció JELZÉS

THEN RESET  STOP_L    'Stop lámpa JEL
  SET       IP_FLAG    'Indítási protokoll FLAG
```

"" Paletta munkadarab számlálás

```
IF      P_SZ      'Paletta munkadarab számláló JELZÉS
AND    N    F0.2    'Edge FLAG

THEN SET      F0.2    'Edge FLAG
INC    CW0      'Jelenlegi darabszám
SET    W_D      'Watchdog időzítő
WITH   60s
SET    T5      'Vákuum watchdog időzítő
WITH   60s

IF      P_FULL    'Paletta telítettség JELZÉS
AND    N    IP_FLAG    'Indítási protokoll FLAG

THEN SET      P1      'Paletta tele lámpa villogás PROGRAM
LOAD    (  CW0      'Jelenlegi darabszám
+      CW1      'Összes darab SZÁMLÁLÓ
TO      CW1      'Összes darab SZÁMLÁLÓ
```

"" Paletta ürítés

```
IF      P1      'Paletta tele lámpa villogás PROGRAM
AND    P_EMPTY    'Paletta ürességet jelző GOMB (?)
AND    PAL_SZ    'Paletta jelenlét szenzor

THEN
  RESET    P1      'Paletta tele lámpa villogás PROGRAM
  SET      P_OK    'Paletta üres JEL a robot felé
  SET      C0      'Cikluson belüli SZÁMLÁLÓ
  SET      IP_FLAG    'Indítási protokoll FLAG

IF      (  CW0      'Jelenlegi darabszám
>      V0      )
AND    N    IP_FLAG    'Indítási protokoll FLAG
THEN
  RESET    P_OK    'Paletta üres JEL a robot felé
```

"" Vészleállítás

```
IF      N    EMSTOP    'Vészleállító GOMB
THEN SET    EMSTOP_L    'Vészleállító lámpa JEL
SET      R_V_STOP    'Robot vészstop JEL
RESET    START_L    'Start lámpa JEL
RESET    STOP_L    'Stop lámpa JEL
RESET    R_START    'Robot start/ Robot üzem PARANCs
RESET    POZ_J    'Munkadarab jelenlét JEL
```

```
RESET    R_STOP    'Robot ciklus vége stop PARANCS
RESET    IP_FLAG    'Indítási protokoll FLAG
```

```
IF        EMSTOP_L    'Vészleállító lámpa JEL
AND       NYUGTA      'Vészstop nyugtázás
THEN
  RESET    EMSTOP_L    'Vészleállító lámpa JEL
  RESET    R_V_STOP    'Robot vészstop JEL
  SET      IP_FLAG     'Indítási protokoll FLAG
  SET      C2          'Hibakód
```

"" Munkadarab pozíció ellenőrzés felvételhez

```
IF        POZ_SZ      'Munkadarab pozíció szenzor JELZÉS
AND       P_RUN       'Palettázás folyamatban JELZÉS
THEN SET   POZ_J      'Munkadarab jelenlét JEL

IF        N POZ_SZ     'Munkadarab pozíció szenzor JELZÉS
THEN RESET POZ_J      'Munkadarab jelenlét JEL
```

"" Robot kommunikáció ellenőrzés

```
IF        R_COMM      'Robot kommunikáció JELZÉS
THEN SET   R_COMM_F    'Robot kommunikáció FLAG

IF        N R_COMM     'Robot kommunikáció JELZÉS
THEN RESET R_COMM_F    'Robot kommunikáció FLAG
```

"" Palettázás futás ellenőrzés

```
IF        P_RUN       'Palettázás folyamatban JELZÉS
THEN SET   P_RUN_F     'Palettázás futás FLAG

IF        N P_RUN      'Palettázás folyamatban JELZÉS
THEN RESET P_RUN_F     'Palettázás futás FLAG
```

"" Paletta telítettség ellenőrzés

```
IF        P_FULL      'Paletta telítettség JELZÉS
THEN SET   P_FULL_F    'Paletta tele FLAG
  SET      T2          'Watchdog időzítő
  WITH     60s

IF        N P_FULL     'Paletta telítettség JELZÉS
THEN RESET P_FULL_F    'Paletta tele FLAG
```

"" Watchdog ellenőrzés és hibakeresés

```
IF        N W_D        'Watchdog időzítő
AND       N IP_FLAG     'Indítási protokoll FLAG
THEN SET   P2          'Hibakeresés PROGRAM
```

```
IF        P_EMPTY     'Paletta ürességet jelző GOMB (?)
```

```

THEN SET      W_D      'Watchdog időzítő
  WITH      60s

"" Vákuum rendszer ellenőrzés

IF          VAC_SZ      'Vákuum szenzor
  AND      N      F0.3      'Edge flag
THEN SET      F0.3      'Edge flag
  SET      T5      'Vákuum watchdog időzítő
  WITH      60s

IF          N      T5      'Vákuum watchdog időzítő
  AND      P_RUN      'Palettázás folyamatban JELZÉS
  AND      N      IP_FLAG      'Indítási protokoll FLAG
THEN LOAD      V6
  TO      CW2      'Hibakód számláló

```

- Paletta lámpa villogás program

```

STEP 10
IF          N      T1      'Villogás időzítő
THEN SET      P_FULL_L      'Paletta tele lámpa
  SET      T1      'Villogás időzítő
  WITH      0.5s

STEP 20
IF          N      T1      'Villogás időzítő
THEN RESET      P_FULL_L      'Paletta tele lámpa
  SET      T1      'Villogás időzítő
  WITH      0.5s
JMP TO 10

```

- Hibakeresés program

```

IF          P_RUN      'Palettázás folyamatban JELZÉS
  AND      N      POZ_SZ      'Munkadarab pozíció szenzor JELZÉS

THEN LOAD      V1
  TO      CW2      'Hibakód számláló

IF          N      R_COMM      'Robot kommunikáció JELZÉS
  AND      IP_FLAG      'Indítási protokoll FLAG
THEN LOAD      V2
  TO      CW2      'Hibakód számláló

IF          R_START      'Robot start/ Robot üzem PARANCS
  AND      ( N P_RUN      'Palettázás folyamatban JELZÉS
  OR      N      P_FULL      ) 'Paletta telítettség JELZÉS

```

```
THEN LOAD      V3
  TO      CW2      'Hibakód számláló
```

```
IF      ST_1      '1-es Ütköző
  AND   N   POZ_SZ      'Munkadarab pozíció szenzor JELZÉS
THEN LOAD      V4
  TO      CW2      'Hibakód számláló
```

```
IF      P_FULL      'Paletta telítettség JELZÉS
  AND   N   PAL_SZ      'Paletta jelenlét szenzor
THEN LOAD      V5
  TO      CW2      'Hibakód számláló
```

- Szállítópálya vezérlés program

```
STEP 10
IF      N   POZ_SZ      'Munkadarab pozíció szenzor JELZÉS
THEN SET      T3      'Szállító időzítő 1
  WITH      1s
```

```
STEP 20
IF      N   T3      'Szállító időzítő 1
  AND   N   POZ_SZ      'Munkadarab pozíció szenzor JELZÉS
THEN SET      ST_1      '1-es Ütköző
```

```
STEP 30
IF      POZ_SZ      'Munkadarab pozíció szenzor JELZÉS
THEN RESET      ST_1      '1-es Ütköző
  SET      ST_2      '2-es Ütköző
  SET      T4      'Szállító időzítő 2
  WITH      2.5s
```

```
STEP 40
IF      N   T4      'Szállító időzítő 2
THEN RESET      ST_2      '2-es Ütköző
  JMP TO 10
```

3. Melléklet

- login.html

```
<!DOCTYPE HTML PUBLIC><html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Távfelügyelet bejelentkezés</title>
<style type="text/css">
form{
    margin:auto;
    width:350px;
    height:350px;
    margin-top: 100px;
    background-color:#444444;
    border-radius: 4px;
    padding: 12px;
}
h4{
font-size:25px;
text-align:center;
height:40px;
border-bottom: 2px solid;
border-bottom-color:rgb(5, 174, 188);
}
input[type=text]{
margin:auto;
margin-top:10px;
margin-right:5px;
margin-bottom:5px;
border: 1px solid red;
background-color:#000000;
color:white;
border-radius: 3px;
border-color:#05AEBC;
width:100%;
height:30px;
}
input[type=password]{
margin:auto;
margin-top:10px;
margin-right:5px;
border: 1px solid red;
```

```
background-color:#000000;
color:white;
border-radius: 3px;
border-color:#05AEBC;
width:100%;
height:30px;
}

input[type=button]{
margin:auto;
margin-top:10px;
margin-left:100px;
padding:8px 25px;
cursor:pointer;
border:none;
background-color:#05AEBC;
transition-duration: 0.3s;
border-radius:3px;
}
input[type=button]:hover{
background-color:#015449;
box-shadow:0 8px 16px 0 rgba(0,0,0,0.2), 0 6px 20px 0 rgba(0,0,0,0.19);
color:#CCCCCC;
}

input[type=button]:focus{
outline: none;
}

label{
font-size:18px;
}

body {
background-color: #CCCCCC;
}

#keszito{
float:right;
margin-top:55px;
border-bottom: 1.5px solid #05AEBC;
}
</style>

</head>
<body>
```

```

<form>
  <h4>Palettázó rendszer felügyelet</h4>
  <label for="felhasznalo">Felhasználónév:</label>
  <input type="text" id="felh"><br>
  <label for="jelszo">Jelszó: </label>
  <input type="password" id="jelsz"><br>
  <input type="button" value="Bejelentkezés" onClick="checkData();" />
  <p id="keszito">Készítette: Samu Bence</p>
</form>

<!--Function to check password the already set user/password is Admin/admin
and User/user-->
<script type="text/javascript">
  function checkData() {
    var definialtJelszo = "admin";
    var definialtFelhasznalo = "Admin";
    var definialtJelszo2 = "user";
    var definialtFelhasznalo2 = "User";
    var jelszo = document.getElementById("jelsz").value;
    var felhasznalo = document.getElementById("felh").value;
    if (jelszo == definialtJelszo && felhasznalo == definialtFelhasznalo)
  {
    window.location="main.html";
  }
  else if (jelszo == definialtJelszo2 && felhasznalo ==
definialtFelhasznalo2) {
    window.location="main_2.html";
  }
  else{
    alert("A felhasználónév vagy jelszó nem megfelelő.");
  }
  }
</script>
</body>
</html>

```

- main.html

```
<html lang="hu">
  <head>
    <title>Távfelügyelet Honlap</title>
    <script lang="JavaScript" src="fst.js"></script>
    <meta http-equiv="refresh" content="3"></meta>
  </head>
  <body style="background-color: #CCCCC; font-family: Georgia, 'Times New
Roman', Times, serif;">
    <h1 style="text-align: center;">Palettázási folyamat távfelügyelet</h1>
    <h2 style="text-align: center;">Pontos idő: <span
id="datumido"></span></h2>
    <script>
      var ip = new Date();
      document.getElementById("datumido").innerHTML =
ip.toLocaleString();
    </script>
    <table style="color: white; text-align: left; font-family: Georgia, 'Times
New Roman', Times, serif; background-color: rgb(62, 60, 182); height: 315px;
width: 1295px;" border="1" cellpadding="2" cellspacing="2">
      <thead>
        <tr style="text-align: center; font-weight: bold; height: 50px;">
          <th colspan="2"><big>A palettázás jelenlegi
állapota</big></th>
        </tr>
      </thead>
      <tbody>
        <tr>
          <td style="text-align: center; font-weight: bold; width:
600px; height: 50px;"><big>Megnevezés</big></td>
          <td style="text-align: center; font-weight: bold; height:
50px;"><big>Állapot</big></td>
        </tr>
        <tr>
          <td style="height: 100px; text-align: center;">Robot
kommunikáció</td>
          <td style="text-align: center;"><script language="javascript">
switch(<FSTCI dmw1>){
  case 0: document.write("A robot nem kommunikál"); break;
  case 1: document.write("A robot kommunikál"); break;
  default: document.write("Kapcsolódás"); break;
}
</script></td>
        </tr>
        <tr>
```

```

        <td style="height: 100px; text-align: center;">Palettázás
ciklus</td>
        <td style="text-align: center;"><script language="javascript">
switch(<FSTCI dmw2>){
case 0: document.write("Várákozik"); break;
case 1: document.write("Folyamatban"); break;
default: document.write("Nincs kapcsolat"); break;
}
        </script></td>
</tr>
<tr>
        <td style="height: 100px; text-align: center;">Palettán
elhelyezett darabok száma</td>
        <td style="text-align: center;"><FSTCI dzw0></td>
</tr>
<tr>
        <td style="height: 100px; text-align: center;">Paletta
telítettsége</td>
        <td style="text-align: center;"><script language="javascript">
switch(<FSTCI dmw3>){
case 0: document.write("Van hely"); break;
case 1: document.write("TELE"); break;
default: document.write("Nincs kapcsolat"); break;
}
        </script></td>
</tr>
<tr>
        <td style="height: 100px; text-align: center;">Összes álpakolt
darab</td>
        <td style="text-align: center;"><FSTCI dzw1></td>
</tr>
<tr>
        <td style="height: 100px; text-align: center;">Watchdog</td>
        <td style="text-align: center;"><script language="javascript">
switch(<FSTCI dmw5>){
case 0: document.write("Nincs hiba"); break;
case 1: document.write("HIBA"); break;
default: document.write("Kapcsolódás"); break;
}
        </script>
</td>
</tr>
<tr>
        <td style="height: 100 px; text-align: center;">Hiba típusa</td>
        <td style="height: 100 px; text-align: center;"><script
language="javascript">
switch(<FSTCI dzw2>){
case 0:document.write("Nincs hiba"); break;

```

```
break;
        case 1:document.wtite("Nincs munkadarab / Szenzor hiba");
        case 2:document.wtite("Robot nem kommunikál"); break;
        case 3:document.wtite("Robotkar hiba"); break;
        case 4:document.wtite("Szállítópálya hiba"); break;
        case 5:document.wtite("Paletta hiba"); break;
        case 6:document.wtite("Vákuum rendszer hiba"); break;
        default: documnet.write("Kapcsolódás"); break;
    }
</script>
</td>
</tr>
</tbody>
</table>
</body>
</html>
```

- main_2.html

```
<html lang="hu">
  <head>
    <title>Távfelügyelet Honlap</title>
    <script lang="JavaScript" src="fst.js"></script>
    <meta http-equiv="refresh" content="3"></meta>
  </head>
  <body style="background-color: #CCCCC; font-family: Georgia, 'Times New
Roman', Times, serif;">
    <h1 style="text-align: center">Palettázási folyamat távfelügyelet</h1>
    <h2 style="text-align: center;">Pontos idő: <span
id="datumido"></span></h2>
    <script>
      var ip = new Date();
      document.getElementById("datumido").innerHTML =
ip.toLocaleString();
    </script>
    <table style="color: white; text-align: left; font-family: Georgia, 'Times
New Roman', Times, serif; background-color: rgb(62, 60, 182); height: 315px;
width: 1295px;" border="1" cellpadding="2" cellspacing="2">
      <thead>
        <tr style="text-align: center; font-weight: bold; height: 50px;">
          <th colspan="2"><big>A palettázás jelenlegi
állapota</big></th>
        </tr>
      </thead>
      <tbody>
        <tr>
          <td style="text-align: center; font-weight: bold; width:
600px; height: 50px;"><big>Megnevezés</big></td>
          <td style="text-align: center; font-weight: bold; height:
50px;"><big>Állapot</big></td>
        </tr>
        <tr>
          <td style="height: 100px; text-align: center;">Palettázás
ciklus</td>
          <td style="text-align: center;"><script language="javascript">
switch(<FSTCI dmw2>){
  case 0: document.write("Várakozik"); break;
  case 1: document.write("Folyamatban"); break;
  default: document.write("Nincs kapcsolat"); break;
}
</script></td>
        </tr>
      </tbody>
    </table>
```

```

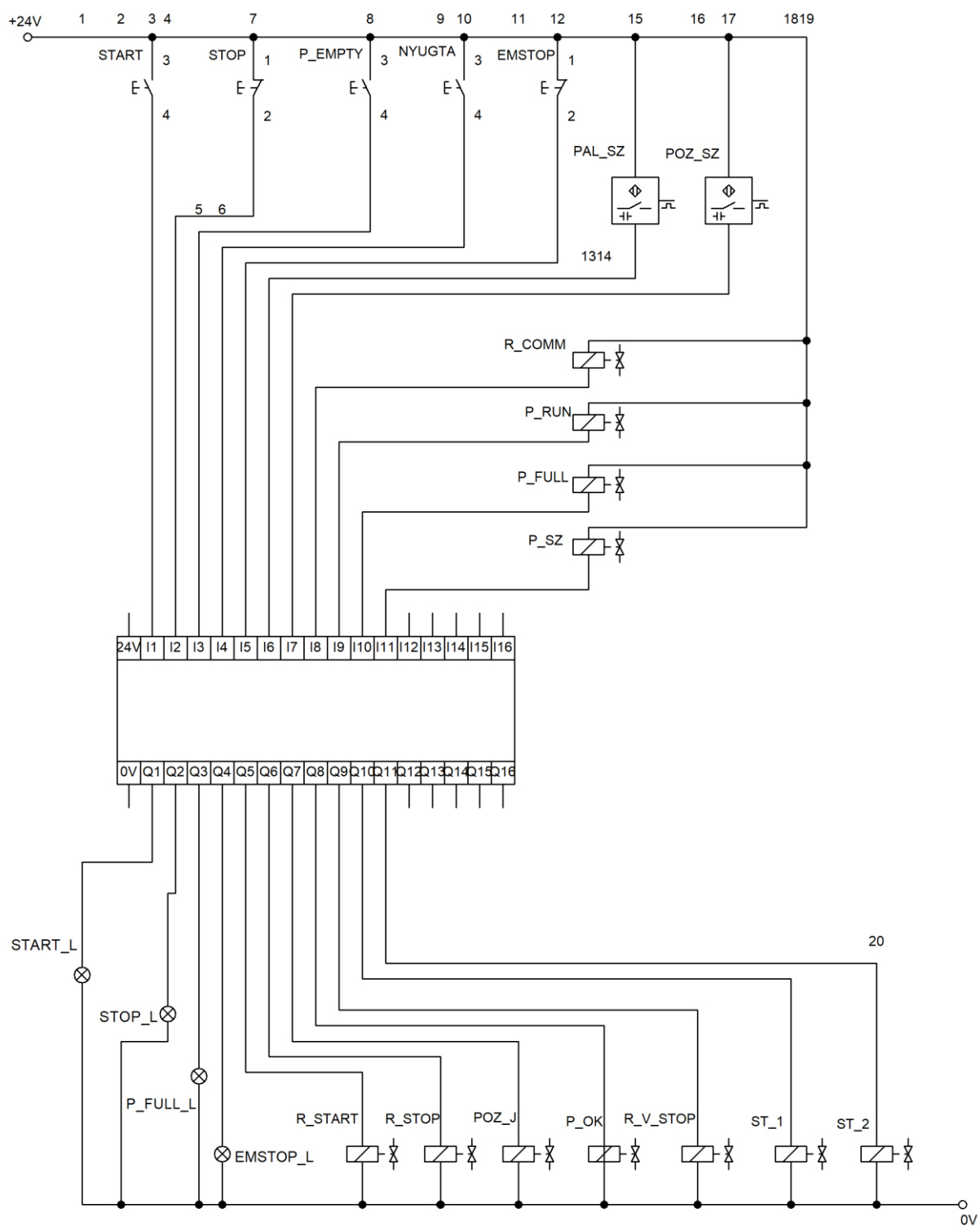
        <td style="height: 100px; text-align: center;">Összes álpakolt
darab</td>
        <td style="text-align: center;"><FSTCI dzw1></td>
    </tr>
    <tr>
        <td style="height: 100px; text-align: center;">Watchdog</td>
        <td style="text-align: center;"><script language="javascript">
switch(<FSTCI dmw5>){
case 0: document.write("Nincs hiba"); break;
case 1: document.write("HIBA"); break;
default: document.write("Kapcsolódás"); break;
}
</script>
</td>
    </tr>
    <tr>
        <td style="height: 100px; text-align: center;">Hiba
típusa</td>
        <td style="height: 100px; text-align: center;"><script
language="javascript">
switch(<FSTCI dzw2>){
case 0:document.write("Nincs hiba"); break;
case 1:document.wtite("Nincs munkadarab / Szenzor hiba");
break;

case 2:document.wtite("Robot nem kommunikál"); break;
case 3:document.wtite("Robotkar hiba"); break;
case 4:document.wtite("Szállítópálya hiba"); break;
case 5:document.wtite("Paletta hiba"); break;
case 6:document.wtite("Vákuum rendszer hiba"); break;
default: documnet.write("Kapcsolódás"); break;
}
</script>
</td>
    </tr>
</tbody>
</table>
</body>
</html>

```

4. Melléklet

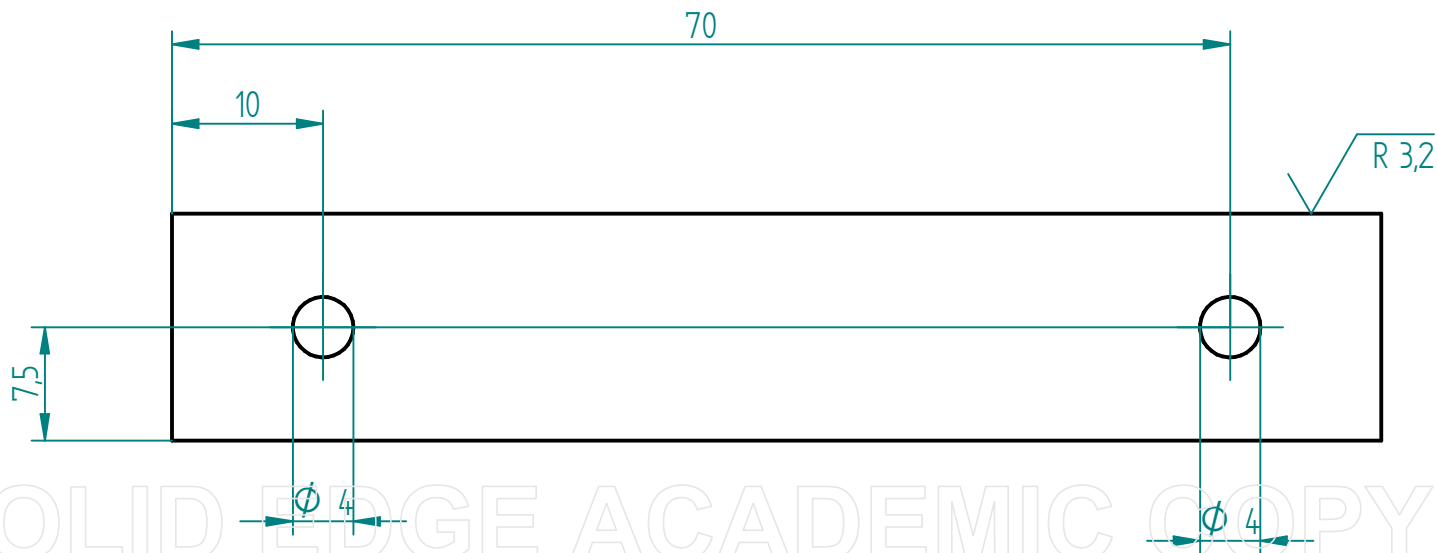
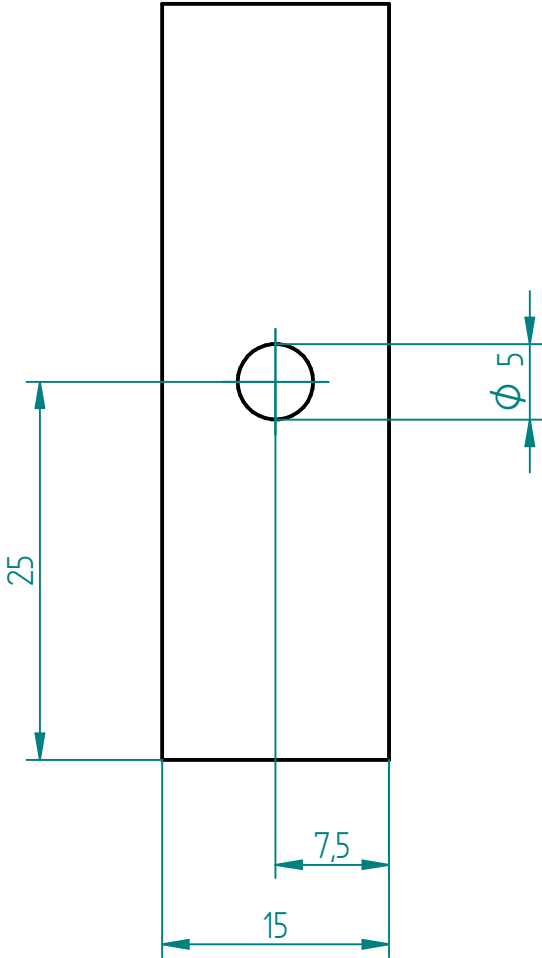
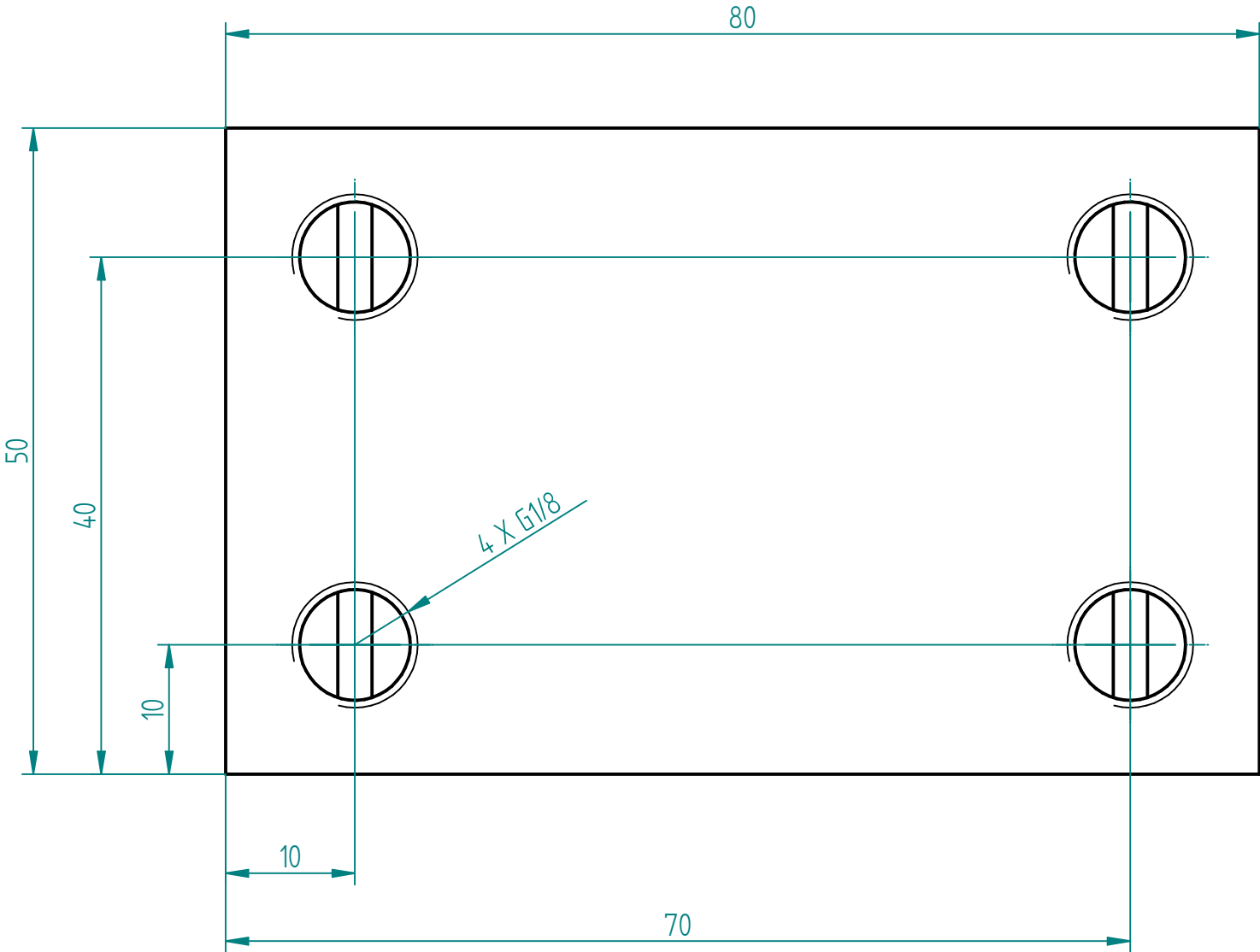
- Elektromos kapcsolás



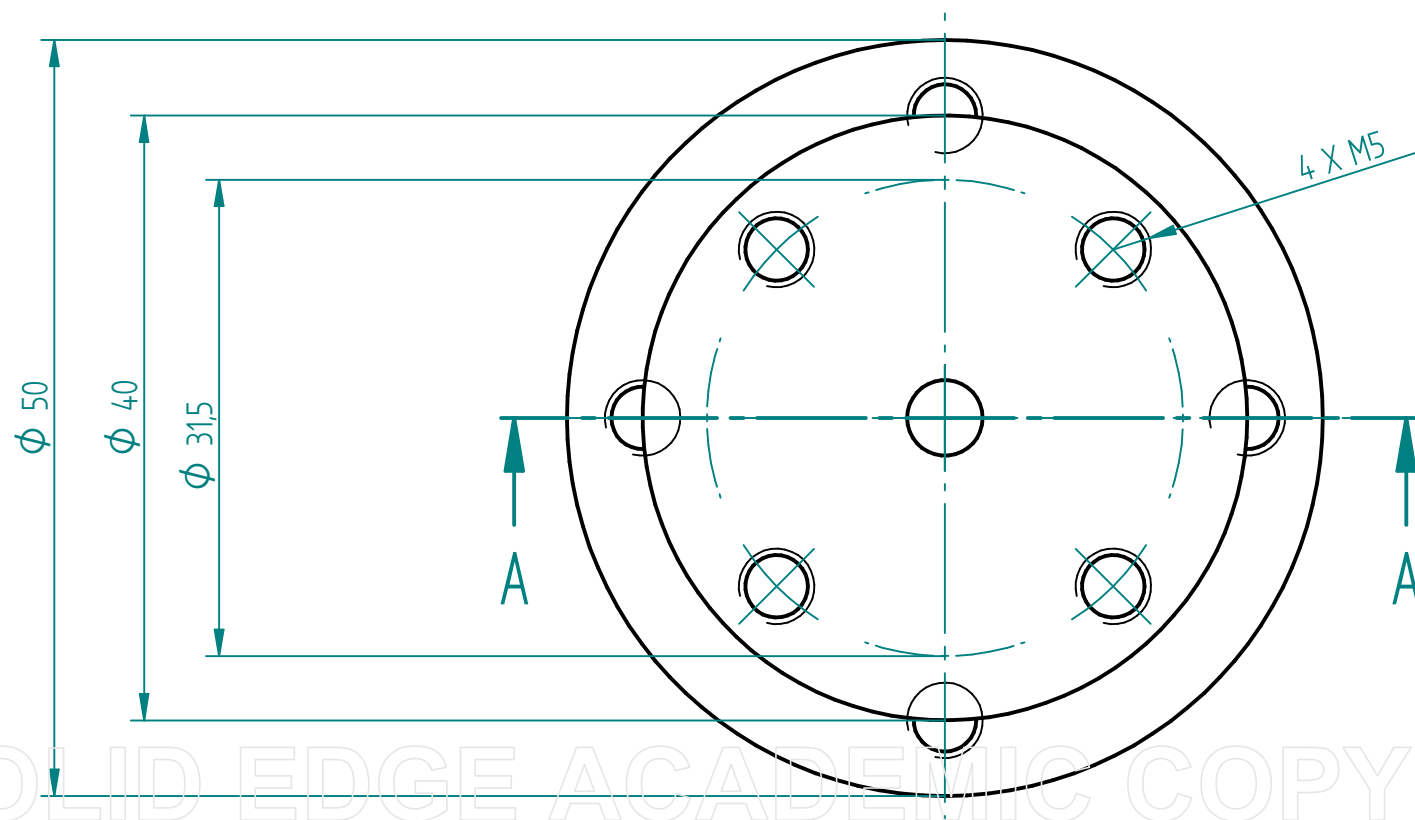
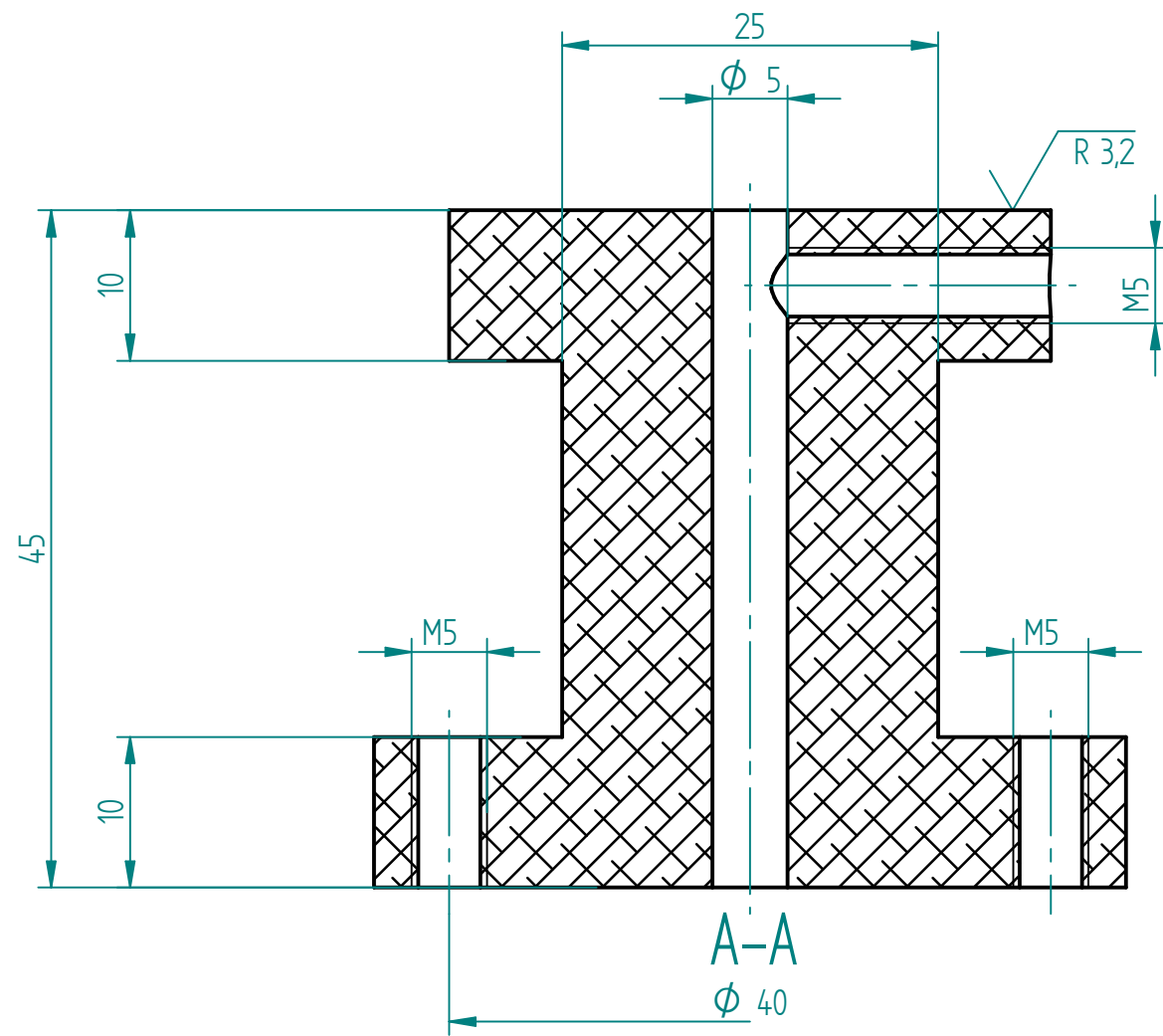
5. Melléklet

Darabjegyzék			
Sorszám	Darab	Megnevezés	Típus
1.	1	Vákuumgenerátor	Festo VADMI-95-P
2.	4	Vákuumkorong	Festo VASB-15-1/8-PUR-B
3.	2	Pneumatikus munkahenger	Festo AEN-20
4.	2	Kapacitív szenzor	Balluff BCS00PU
5.	1	Szállítópálya	Tech-Con D32
6.	2	Pneumatikus útváltó szelep	Festo VSNC-FK-M32C-RD-G18-1C1-S
7.	1	Vákuum elosztó elem	
8.	2	Vákuum tartó elem	
9.	1	Robotkar	Mitsubishi RV-2AJ
10.	1	Robot vezérlő	Mitsubishi CR1-571
11.	1	Programozható logikai egység	Festo FEC-CPX
12.	1	I/O modul	Festo CPX-E-16DI
13.	2	Fojtó visszacsapó szelep	Festo VFOE-LE-T-M5-Q4
14.	1	Levegő-előkészítő egység	Festo MS4-EM1FR-1/4-D6-C-P-VC-A8-WPE-B

Revíziótörténet			
REV	Leírás	Dátum	Jóváhagyva



	Név	Dátum	Magyar Agrár- és Élettudományi Egyetem		
TERVEZTE	Samu Bence	2023/10/25			
Ellenorizve					
Anyag:	Aluminium, 1060		Megnevezés Vákuum korong tartó		
ELTÉRO RENDELKEZÉS HIÁNYÁBAN MÉRETEK MILLIMÉTERBEN ÉRTENDOK			Lap m:	RAJZSZÁM	REV
			A3	SB-SZAK-0002	
			Fájlnev Vacuum_holder.dft		
			Mérete: 2:1	Tömeg: 0,15 kg	Lapok 1 / 1



Revíziótörténet			
REV	Leírás	Dátum	Jóváhagyva

	Név	Dátum	Magyar Agrár- és Élettudományi Egyetem		
TERVEZTE	Samu Bence	2023/10/25			
Ellenorizve					
Anyag:	Aluminium, 1060		Megnevezés Vákuum tartó kapcsoló elem		
ELTÉRO RENDELKEZÉS HIÁNYÁBAN MÉRETEK MILLIMÉTERBEN ÉRTENDOK			Lap m.: A3	RAJZSZÁM SB-SZAK-0001	REV
			Fájlnév Vacuum_joint.dft		
			Méreta: 2:1	Tömeg: 0,115 kg	Lapok 1 / 1