

SZAKDOLGOZAT

Kardos Márton

2023

i



Magyar Agrár- és Élettudományi Egyetem

Szent István Campus

Műszaki Intézet

Gépipari automatizálási szakmérnök

**Mesterséges intelligencia alapú
gépfelügyeleti rendszer fejlesztése**

Belső konzulens: Dr. Jánosi László

Prof. Emeritus

**Belső konzulens
intézete/tanszéke:**

Műszaki intézet

Külső konzulens:

Albert Viktor

Alkalmazásmérnök

Készítette:

Kardos Márton

2023

**MŰSZAKI INTÉZET
GÉPIPARI AUTOMATIZÁLÁSI SZAKMÉRNÖK**

DIPLOMADOLGOZAT
feladatlap

Kardos Márton (MSJ3BZ)

részére

A diplomadolgozat címe:

Mesterséges intelligencia alapú gépfelügyeleti rendszer fejlesztése

Feladatkiírás:

Bevezetés, Szakirodalom feldolgozása, Probléma bemutatása, Mesterséges intelligencia alapú gépfelügyeleti rendszer fejlesztése, Tovább lépési lehetőségek, Gazdasági számítás, Összefoglalás

Közreműködő tanszék: Mechatronika

Külső konzulens: *Albert Viktor, alkalmazásmérnök, FESTO Kft.*

Belső konzulens: *Dr. Jánosi László, professor emeritus, MATE, Műszaki Intézet*

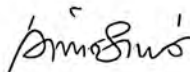
Beadási határidő: 2023. november 06.

Gödöllő, 2023. szeptember 04.

Jóváhagyom

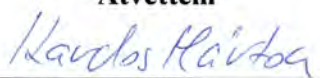


(tanszékvezető)



(szakfelelős)

Átvettem



(hallgató)

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

Gödöllő, 2023. 10. hó 30. nap



(külső konzulens)

TARTALOMJEGYZÉK

1. Bevezetés	1
2. Irodalmi áttekintés	2
2.1. Manipulátorok, ipari robotok	2
2.2. Szenzortechnika	5
2.2.1. Szenzorok és csoportosításuk.....	6
2.2.2. A szortírozó berendezésben alkalmazott szenzorok	7
2.3. PLC	8
2.3.1. A PLC-k felépítése.....	9
2.3.2. PLC-k programozásának fejlődése	10
2.4. Terepi busz rendszerek.....	10
2.5. Szervo és léptető motorok	12
2.5.1. Léptető motorok.....	12
2.6. Ipar 4.0 és kapcsolódó technológiái	13
2.6.1. Az ipari forradalmak és vívmányaik.....	13
2.7. AI alapú gépfelügyeleti rendszerek.....	14
2.7.1. Ipari Mesterséges intelligencia.....	14
2.7.2. A mesterséges intelligencia típusai	15
2.7.3. Gép állapotfelügyeleti rendszer	16
2.7.3.1. Állapotfelügyeleti rendszer működése.....	16
2.7.4. Karbantartás és mesterséges intelligencia kapcsolata	17
2.7.5. Az állapotfelügyelethez használt leggyakoribb szenzorok	18
2.7.6. Festo AX rendszere	18
2.8. Node-RED böngésző alapú grafikus programozás	21
2.9. Az MQTT protokoll	22
2.10. OPC-UA szabvány	23
3. Mesterséges intelligencia alapú gépfelügyeleti rendszer fejlesztése	24
3.1. A válogatási feladat leírása	25

3.2. A belső kommunikációs hálózat felépítése	25
3.3. Az Automation Experience architektúrája és működése	26
3.4. Egyrendeléses gyártás módosítása folyamatos üzemre.....	27
3.5. OPC-UA szerver és kliens összekötése.....	31
3.6. Rendelkezésre álló adatok az OPC-UA-n	31
3.7. A Festo AX futtatási környezete	32
3.8. A Node-RED csatlakoztatási felület	33
3.9. A szortírozó manipulátor adatainak beolvasása a Festo AX rendszerbe.	37
3.10. Az AX irányítópult létrehozása és beállítása	38
3.11. Mesterséges anomália generálása a manipulátoron	38
3.12. Sikertelen modellgenerálás és anomália érzékelés	39
3.13. Várt eredmények és konklúzió.....	40
3.14. Gazdasági számítás	42
4. Összefoglalás	43
5. Summary.....	45
6. Irodalomjegyzék	47

1. BEVEZETÉS

A Fehér Ház legutóbbi, a mesterséges intelligenciáról (AI) szóló jelentése kiemeli az AI jelentőségét, valamint egy világos ütemterv és stratégiai befektetés szükségességét ezen a területen. Ahogy a mesterséges intelligencia a tudományos-fantasztikus irodalomból a világot megváltoztató technológiák élvonalába emelkedik, sürgősen szükség van a mesterséges intelligencia szisztematikus fejlesztésére és megvalósítására, hogy meglássuk valódi hatását az ipari rendszerek következő generációjában, nevezetesen az Ipar 4.0-ban. (Lee, 2016)

Szakedolgozatom célja egy az ipar 4.0.-ba illeszkedő technológia megismerése és bemutatása. Ezt egy preparált szortírozó manipulátor segítségével fogom megtenni. Valamint egy mesterséges intelligencián alapuló gépfelügyeleti rendszert fogok tesztelni.

A manipulátor a Festo Kft. bemutató eszköze, mely a cég budapesti telephelyének váróaulájában található. Választásom azért esett a berendezésre, mert célom az elektromos hajtások, a PLC vezérlés, ipari hardver kommunikáció és a mesterséges intelligencián alapú ipari gépfelügyeleti rendszer megismerése, megértése, valamint fejlesztése. A gép kiváló lehetőségeket biztosít erre a feladatra.

Feladatom a berendezés állapotának felmérése, alkatrészek és kapcsolt rendszerek azonosítása lesz - ideértve az elektromos hajtásokat, a portálrendszert, a szenzorokat, a PLC-t, kommunikációs hálózatot, pneumatikus elemeket stb. Majd a Festo AX mesterséges intelligencián alapuló gépfelügyeleti rendszert összekapcsolom a berendezéssel, hogy megfigyelhető legyen a berendezés működése. Majd a mesterséges hibák generálásával próbálom reakcióra bírni a mesterséges intelligenciát. A Festo AX egy okos gyártást támogató rendszer, amely 2021-ben debütált a nagyközönségnek, így egy friss jövőbe mutató technológia, mellyel egyre gyakrabban találkozhatunk az ipar 4.0 terjeszkedésével.

2. IRODALMI ÁTTEKINTÉS

Napjainkban az automatizálás az egyik igen gyakran használt kifejezés mind a lakossági, mind a gazdasági, mind a gyártási, termelési folyamatokban. Az iparban régóta örvend nagy sikernek, de a háztartásokban is egyre gyakrabban alkalmazzák valamilyen formában. Segítségével nagymértékben növelhető a folyamatok sebessége, minősége és pontossága az emberi tényező elhagyásával. A gazdasági növekedés eszközének is tekinthető, mivel egyre újabb és újabb berendezések jelennek meg az iparban, amelyek megkönnyítik az ember munkáját, de magasabb képzettségi szintet is elvárnak a használóiktól. Később ezek a háztartásokba is beszivárognak egyszerűsített formában, így segítve a technológia még szélesebb körű felhasználását. (Kulcsár at al.,2012)

„Az automatizálás a termelési folyamat fejlődésének az a szakasza, amely mentesíti az embert nemcsak a fizikai munka, hanem a termelésirányító tevékenység végzése alól is. Az automata görög latin szó, önműködő irányítást végző berendezést jelent” (Kulcsár at al.,2012)

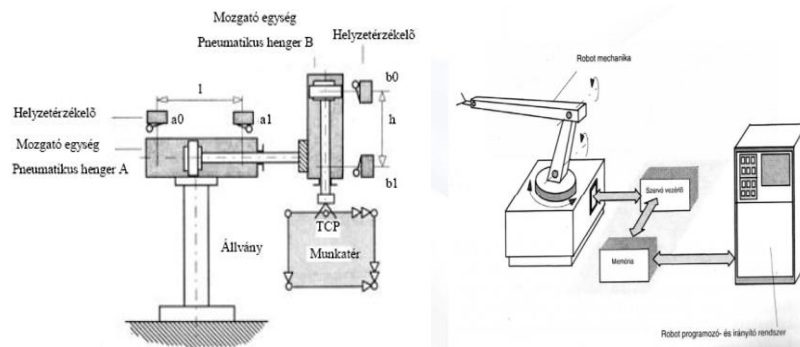
A gyártási folyamatok automatizálásában az 1950-es évek jelentettek fordulópontot. Megjelentek a számítógép irányítású szerszámgépek, megnevezésük a „Numerical Control¹” kezdőbetűiről NC gépek néven vált ismertté. Az NC gépek minőségi fejlődésével a termelékenység elért egy olyan szintet, amelyet tovább növelni csak a folyamatokban résztvevő kiszolgálógépek (anyagmozgatógépek) automatizálásával, illetőleg magának a teljes gyártási folyamatnak az optimális irányításával lehetett. (Kulcsár at al.,2012)

2.1. Manipulátorok, ipari robotok

A robotok és manipulátorok témakörét az ISO 8873:2021 szabvány alapján vizsgáltam meg.

Manipulátor fogalma: „A manipulátor egy olyan mechanizmus, amelyben a szegmensek forgó vagy transzlációs ízülettal kapcsolódnak egymáshoz. A manipulátor definíció szerint nem tartalmazza az end-effektort.” (ISO 8373:2021)

¹ A Numerical Control angol szó, jelentése: számjegyevezérlés.

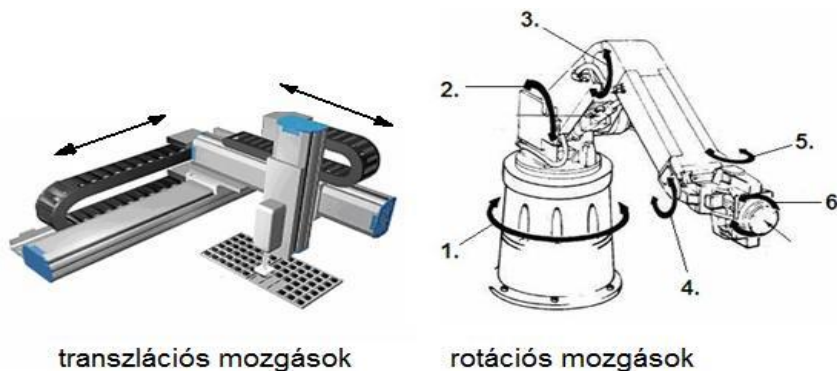


1. ábra. Egy helyező manipulátor és ipari robot elvi vázlata

A mozgástengelyeik lehetnek:

- Transzlációs (T - egyenes vonal mentén elmozduló), és/vagy
- Rotációs (R - forgó) mozgástengely.

Ipari robot fogalma: „A robot olyan programozott működtető mechanizmus, amely bizonyos fokú autonómiával rendelkezik a mozgás, a manipuláció vagy a pozicionálás végrehajtásához. A vezérlőrendszert is magukba foglalják a definíció szerint. (ISO 8373:2021)



2. ábra. Transzlációs és rotációs mozgások (Kodácsy at al., 2011)

Jellemzőik:

- Autonóm: Az a tulajdonság, mely szerint a robot alkalmas feladatok végrehajtására emberi beavatkozás nélkül.
- Újraprogramozható: A mozgások vagy más kiegészítő funkciók megváltoztathatóak a robot szerkezeti átalakítása nélkül.
- Többcélú: Több(féle) feladat ellátásra alkalmas szerkezeti változtatás nélkül (Az effektor és egyéb külső eszközök megváltoztathatók, de a robotot magát nem kell módosítani). (Mayerné, 2022)

Az elvégzendő feladat alapján anyagmozgató és technológiai műveleteket végző robotokat különböztetünk meg.

Anyagmozgató:

- Technológiai gépeket kiszolgálók,
- Rakodó robotok.

Technológiai műveleteket végzők:

- Hegesztőrobotok,
- Festőrobotok,
- Szerelőrobotok.

Az őket működtető energia alapján **pneumatikus, hidraulikus, elektromos** fajtájuk létezik.

Ezen kívül még számos módon csoportosíthatóak az ipari robotok.

Ezek lehetnek:

- Tengelyszám,
- Mozgatható tömeg,
- Betanítási mód,
- Fejlettségi mód alapján.

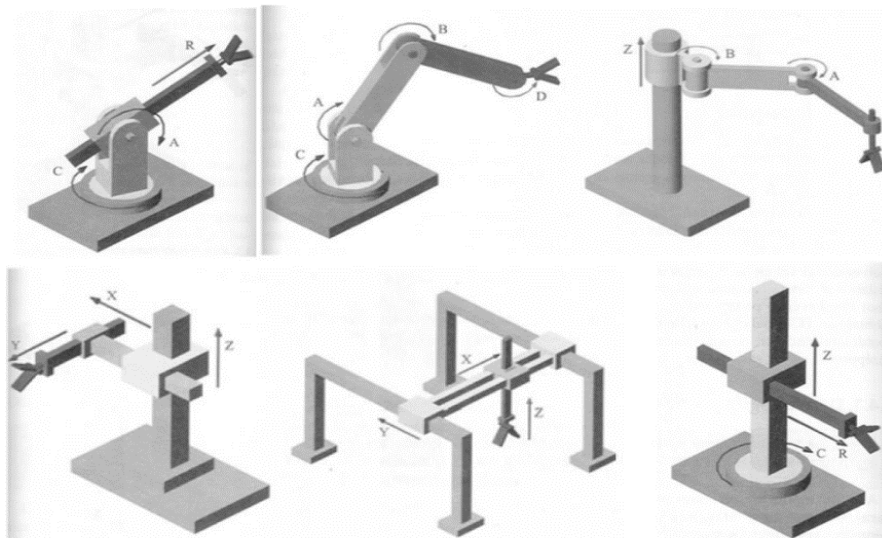
Minden robot rendelkezik úgynevezett „világ-koordináta” rendszerrel ezek lehetnek derékszögű, hengeres, gömbi, illetve csuklókaros rendszerűek.

Derékszögű koordináta rendszerű (TTT) robotok karjait egyenes vonalú (transzlációs), többnyire merőleges tengely mentén mozgatjuk. Kivitele lehet a hasábrobot (nincs holttér) és portálrobot (nagy munkatér, több szerszámgépet is kiszolgálhat).

Henger koordináta rendszerű (RTT) robotok egy rotációs és két transzlációs tengely mentén mozgó, egyszerű vezérlésűek, amelyek az egyik legrégebben kifejlesztett típusok. Szűk hozzáférések esetén előnyös a présgépek, forgácsoló szerszámgépekhez.

Gömbi koordináta rendszerű (RRT) egy transzlációs, két rotációs tengely mentén mozgó, gömb mozgástérrel rendelkező kevésbé elterjedt robot típus.

Csuklókaros (RRR+) robotok csak rotációs tengelyekkel rendelkeznek, a legelterjedtebb típus, nagy pontossága és munkatere miatt. hátránya a kis terhelhetőség és a bonyolult programozás. (Siciliano et al., 2016)



3. ábra. Robot típusok (Miskolci, 2014)

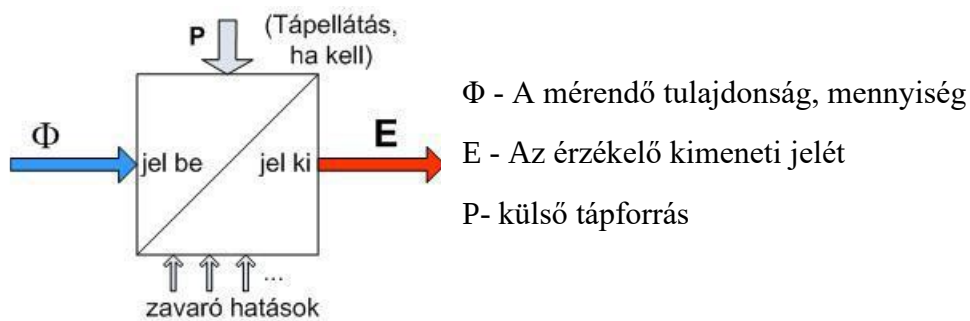
2.2. Szenzortechnika

„Általános értelemben a szenzor egy olyan eszköz, ami valamely természeti mennyiséget műszaki, ill. emberi környezetben jobban kezelhető, jobban kiértékelhető jellé alakít át.” (Bánki at al., 2012) Célunk az, hogy jobban kiértékelhető, könnyen alakítható mennyiséget kapjunk az átalakítás után. A mérendő jel alapján különböztethetünk meg számos típust. Ezek az alábbiak lehetnek:

Napjaikban alkalmazott szenzorok kimenetén jellemzően elektromos jel távozik, mivel ezek széles tartományban erősíthetőek, lehetséges a távmérés, de ezek mellett számos más előnnyel rendelkezik. Lehet természetesen más típusú jel, akár hidraulikus, pneumatikus vagy bármilyen mérhető mennyiség. A 4. ábrán a szenzorok funkcionális modelljét láthatjuk a szokott (szakirodalomban használt) jelölésrendszert alkalmazva. Továbbá a szenzorok lehetnek még:

- passzív, és
- aktív szenzorok.

Passzív érzékelők esetén a működéséhez szükséges segédenergiát betáplálni szükséges, míg az aktív érzékelők az energiát a mért mennyiség rendszeréből nyerik. (Bánki at al., 2012)



4. ábra. Szezor funkcinális modellje (Bánki at al., 2012)

2.2.1. SZENZOROK ÉS CSOPORTOSÍTÁSUK

A szenzorokat számos módon csoportosíthatjuk még. Ezek tárháza szinte végtelen és nehezen standardizálható, ellenben bemutatok néhány csoportosítási módot. Az egyik ilyen, amit gyakran használnak az az energia szemléletű modell. Melynek típusai:

- villamos energia,
- sugárzási energia,
- hőenergia,
- kémiai energia.
- mágneses energia,
- mechanikai energia.

Az előző csoportosításhoz képest talán egyszerűbb és célszerűbb, ha kimeneti jellemzőik alapján osztályozzuk őket. Így az alábbi típusok léteznek:

- feszültség, feszültségváltozás,
- áram, áramváltozás,
- induktivitás változás,
- kapacitásváltozás,
- ellenállásváltozás,
- töltésfelhalmozódás, feltöltöttség változás. (Bánki at al., 2012)

2.2.2. A SZORTÍROZÓ BERENDEZÉSBEN ALKALMAZOTT SZENZOROK

Az ipari szenzorok legsűrűbben használt csoportja az objektumok közelségének érzékelésére szolgál. A tárgyak közelsége mellett akár folyadékszintet is lehet vele érzékelni. A szenzorok ilyen fajtáját a szakirodalom többféle néven is illeti, de térbeli pozíciót, távolságot nem képes érzékelni. A jelenlét, avagy közelségérzékelő megnevezés a legmegfelelőbb. (Bánki at al., 2012)

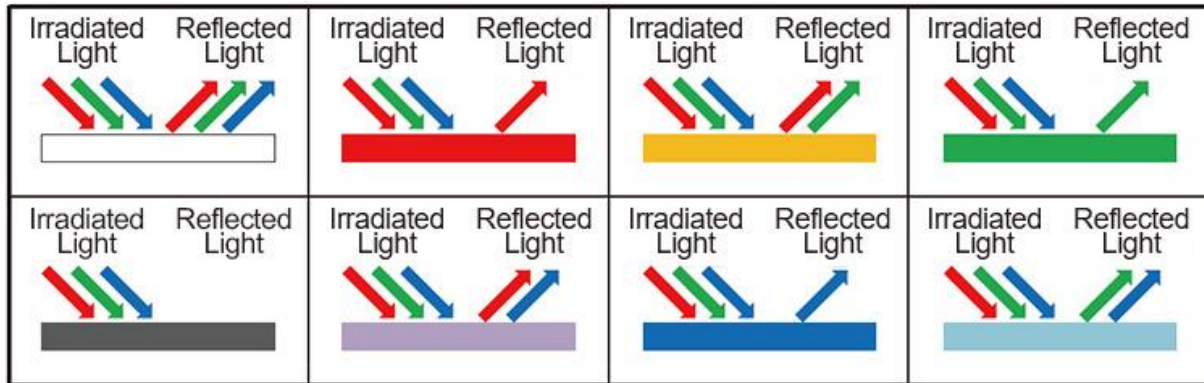
A mágneses közelségérzékelő egy érintésmentes közelségérzékelő, és mágneses tárgyak észlelésére használható. Az ilyen típusú közelségérzékelőket reed érintkezőknek is nevezik. Ez a mágneses tér torzítását használja fel a működéséhez, abban az esetben, ha egy ferromágneses anyag kerül az érzékelő tartományába, akkor a mágneses tér megváltozik, és jelet kap az érzékelő. A mágneses tér számos nem mágneses anyagon áthaladhat, így a kapcsolási folyamat a céltárgy közvetlen expozíciója nélkül is elvégezhető. A mágneses vezető segítségével a mágneses tér nagy távolságokra továbbítható, és ennek köszönhetően a jel elvihető a magas hőmérsékletű területekről. Az ilyen típusú érzékelőknek nincs elektromos zajhatása, és AC vagy DC hálózaton is működhet. Ennek az eszköznek ez az érzékelési távolsága bizonyos tényezőktől, például a hőmérséklettől, az érzékelő objektumtól, a környező tárgytól és az érzékelőtől való felszerelési távolságtól függően változhat. (Krámli, 2008)



5. ábra Magnetoinduktív és optikai közelség érzékelők

Az optikai elven működő érzékelők optikai és elektromos elven működő eszközök kombinációja, mely tárgyak eszközök jelenlét érzékelésére alkalmas. Fényforrásként általában valamilyen fényt kibocsátó diódát (LED-et) használnak, mivel hosszú élettartammal rendelkeznek és kis helyen elférnek. Általában vörös fényt alkalmaznak, mert ez szabad szemmel is látható, és így könnyebb beállítani az érzékelőket. Infravörös fényt ott érdemes használni, ahol nagyobb távolságokat kell áthidalni. Az optikai közelség érzékelők speciális változata a színérzékelő. Ha egy tárgyat RGB-komponenseket (pirosat, zöldet, kékét) tartalmazó fénnel sugároznak be, a visszavert fény színe

az objektum színétől függően változik. Például, ha az objektum piros, a visszavert fénykomponens vörös színű lesz. Sárga tárgy esetén a visszavert fény piros és zöld lesz, és ha az objektum fehér, mindhárom összetevő visszaverődik. Ily módon a tárgy színét a visszavert fényben lévő színösszetevők (RGB) arányából határozzák meg. A színérzékelés folyamata a 6. ábrán látható (Rohm, 2023)



6. ábra Színérzékelés folyamata (Rohm, 2023)

2.3. PLC

Az amerikai autóipar az igen gyakori modellváltások miatt olyan rendszer fejlesztését kezdeményezte, ami eddigi vezérlőrendszer újravezérlés helyett szoftveres úton módosítja programját. A rendszerrel szemben támasztott követelmények a következők voltak 1969-ben a megrendelő General Motors Hydramatic által:

- A vezérlőnek egyszerűnek kell lennie és igény szerint bármikor új programot lehessen vele készíteni.
- Galvanikusan leválasztott I/O egységeket tartalmazzon (24 V DC; 240 V AC),
- A vezérlőnek biztonságosabbnak kell lennie, mint az elektromechanikus megoldás, és ár-érték arányban meg kell előzni a huzalozott kivitelűeket.



7. ábra: Az első PLC és tervezői

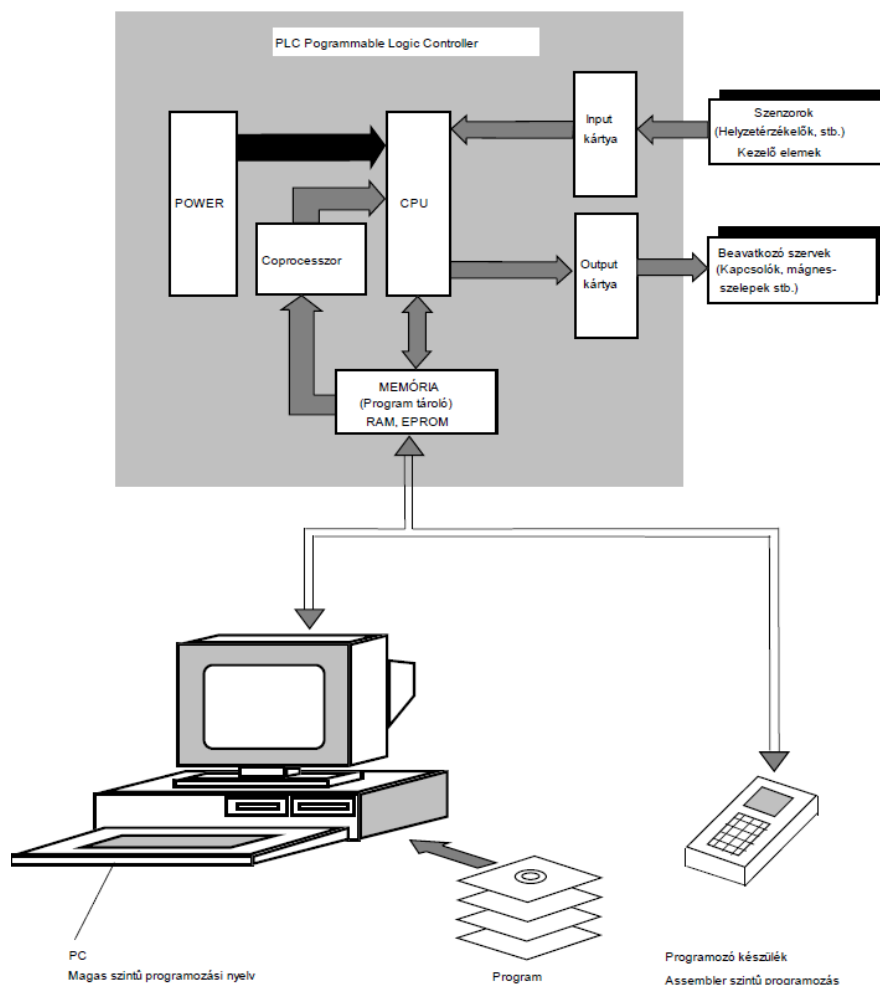
A tervezők a vezérlőberendezésnek a controller nevet adták, mivel az elektronikai és műszaki szakembereket abban az időben elriasztotta volna a computer elnevezés. Az eszköz attól tűnt ki társaitól, hogy a letölthető létra (angol megnevezéssel: ladder logic, német megnevezéssel:

Kontaktpán) program segítségével lehetett módosítani az áramköröket. A logikája az áramút terveket követte. Később a controller szó kiegészült a programmable logic jelzővel a program módosíthatósága miatt, így alakult ki a ma is használt PLC – Programmable Logic Controller. A PLC vezérlés a mai napig az ipar és az automatizálás fontos eleme, megbízhatósága és relatív olcsó ára miatt. (Kulcsár at al., 2012)

2.3.1. A PLC-K FELÉPÍTÉSE

A PLC-k nem összekeverendők a mikrovezérlőkkel, a PLC egy bonyolult programozható eszköz mely rendelkezik:

- mikroprocesszorral (CPU),
- aritmetikai coprocesszorral,
- I/O portokkal,
- Kommunikációs csatlakozókkal



8. ábra: PLC-k elvi felépítése (Kulcsár at al., 2012)

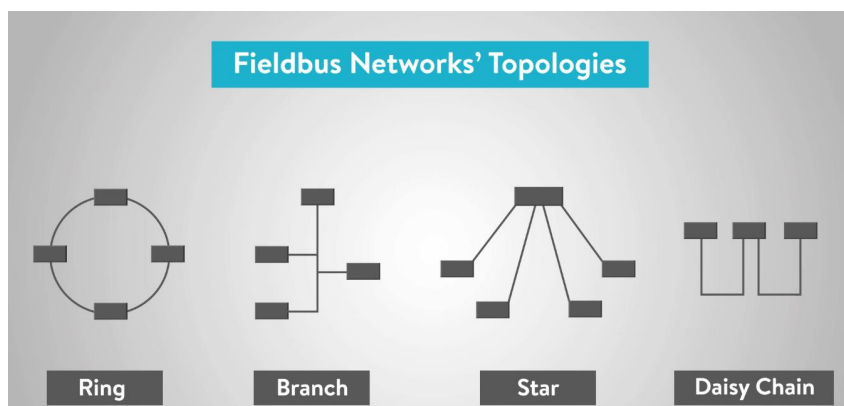
A PLC-k típusait tekintve 2 kivitel különböztethetünk meg, a kompakt és a moduláris kivitel. A kompakt kivitel egy zárt egység. Integrálva van a tápegység, a kommunikációs csatlakozók, valamint a be és kimeneti portok. A portok száma bővíthető bizonyos típusok esetében. A moduláris kivitel ezzel szemben modulokból épül fel. Lehetséges tápegység választás a CPU-hoz, több be- és kimeneti modul csatlakoztatható rá, illetve kommunikációs portok is illeszthetők. Ezek a kivitelek jobban tudnak alkalmazkodni a változó igényekhez a vezérlési feladatok változása esetén. A nagyobb teljesítményű PLC-kre jellemző inkább a moduláris felépítés. (Kulcsár at al., 2012)

2.3.2. PLC-K PROGRAMOZÁSÁNAK FEJLŐDÉSE

A PLC-n két programtípus futtat, az alapprogram vagy más néven operáció rendszer és a felhasználói program. Az operációs rendszer gyárilag meghatározott előre telepített program, nem módosítható. A felhasználói programnyelveket két csoportba sorolhatjuk, a szöveges és grafikus nyelvek. Ezek a programok manapság kizárólag magas szintű számítógépeken készülnek, programozási szoftverek segítségével. A szöveges programnyelvek tovább bonthatóak utasításlista (IL, AWL) típusú és strukturált szöveg (ST) típusú programozásra. Egyre nagyobb jelentőségük a strukturált szöveges verzióknak, mert ezek hasonlítanak a magas szintű programnyelvekre a leginkább, mint a C vagy Python. A grafikus nyelvek a létradiagram (LD, KOP), a funkcióblokk (FB) és a sorrendi folyamatábra (SFC). A PLC vezérlésnél választható a ciklus vagy lépésenkénti vezérlés. (Kulcsár at al., 2012)

2.4. Terepi busz rendszerek

A terepi busz valójában nem egy dolog, hanem inkább a dolgok gyűjteménye. A terepi busz az ipari színtéren használt protokollok csoportja. A terepi busz protokollokat IEC61158

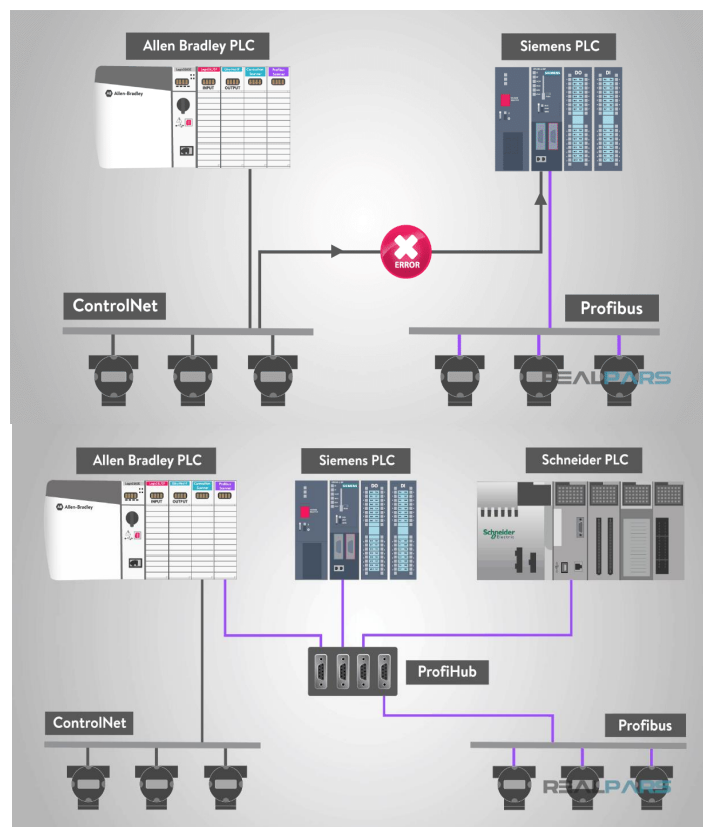


9. ábra: Terepi busz kapcsolási topológiák (RealPars, 2023)

szabvány szerint szabványosították. Alapvetően a terepi busz olyan hálózaton működik, amely lehetővé teszi a különböző kapcsolási topológiákat, például a gyűrűt, az elágazást, a csillagot és a láncot.

A terepi busz protokollok előtt az ipari vezérlőrendszereket RS232 soros kommunikációval kötötték össze. Mint tudjuk, a soros kommunikáció csak két eszköz számára tette lehetővé a kommunikációt. Míg manapság a terepi busz-kapcsolatok jobban hasonlítanak a tipikus Ethernet-kapcsolatokhoz, ahol több helyszíni eszköz csatlakoztatható egyetlen csatlakozási ponthoz, amely aztán a vezérlőhöz csatlakozna.

A terepi busz azonban önmagában nem egy kapcsolattípus, hanem egy leírás, amelyet protokollok csoportjának jelzésére használnak. A csoportban számos protokoll található, például ControlNet az Allen Bradley családban, Modbus, Profibus, EtherCAT, HART, CIP és még sok más.



10. ábra: Kommunikációs lehetőségekre péda (RealPars, 2023)

A terepi busz elsődleges előnye a helyszínen csatlakoztatott eszközök. Vegyük például, hogy több eszközünk van, amelyek meglehetősen távol vannak a vezérlőtől. A terepi buszra csatlakoztatott eszközök, például érzékelők, motorok, lámpák, kapcsolók, egy I/O adatblokkhoz vannak kötve, amely azután egy terepi elosztó eszközhöz csatlakozik, ami egy terepi busz tápegységhez, majd végül a PLC-hez kapcsolódik. A terepi busz eszközök kapcsolata, maguk a terepi

eszközök kivételével, egykábeles csatlakozások. Ez a csatlakozási séma a használt terepi busz protokolltól függően változhat. Az ilyen típusú eszközcsatlakozás megspórolja a felesleges vezetékezést, amelynek egészen a vezérlőig vissza kellene mennie. Az eszközök ilyen típusú csatlakoztatásának nagy előnye van, ha a távolság problémát jelent. (Rostan, 2014)

A vezérlőrendszernek rendelkeznie kell egy kommunikációs modullal, amely képes lesz kommunikációt létesíteni a terepi eszközökkel. Ez nem jelenti azt, hogy minden protokoll egy adott gyártónak van dedikálva, és csatlakoznia kell a védett vezérlőhöz. Például a Profibus egy széles körben használt protokoll, amely csatlakozhat Allen Bradley-hez, Siemenshez, Modiconhoz stb. Ugyanez vonatkozik a Modbus univerzális protokolljára is. Egyes protokollok azonban nem működnek jól másokkal. A ControlNet olyan példa, amely nem integrálható Siemens processzorba. (RealPars, 2023)

2.5. Szervo és léptető motorok

A léptető- és szervomotorok egyaránt népszerű választások különféle iparágakban, például a robotikában, az automatizálásban és a gyártásban. Ezek a motorok nagyon eltérően működnek, és határozott előnyökkel és hátrányokkal rendelkeznek. A különbségek megértése döntő fontosságú az adott projekthez megfelelő motor kiválasztásához. (Athani, 1997)

2.5.1. LÉPTETŐ MOTOROK

A léptetőmotor egy kefe nélküli egyenáramú villanymotor, amely a teljes fordulatot több egyenlő lépésre vagy lépésekre osztja. A motor helyzete precízen szabályozható visszacsatoló mechanizmus nélkül, mivel a motor diszkrét lépésekben mozog a vezérlő által biztosított bemeneti impulzusok alapján.

A léptetőmotorokat általában olyan alkalmazásokban használják, amelyek a forgás és a pozícionálás precíz vezérlését igénylik, például CNC-gépekben, 3D nyomtatókban és robotikában.

A léptetőmotor több kulcsfontosságú alkatrészből áll, amelyek együtt alakítják az elektromos energiát mechanikus mozgássá. A fő alkatrészek közé tartozik a forgórész, az állórész és a motor tekercselése. A rotor a léptetőmotor forgó része, amely általában egy tengelyből és állandó mágnesekből vagy lágy mágneses anyagból áll. Az összetétel a léptetőmotor típusától függ. Az állandó mágneses léptetőmotorokban a forgórész egy vagy több állandó mágneset tartalmaz, amelyek mágneses teret hoznak létre. Az állórész a motor álló része, amely a motor kerülete

mentén elhelyezett elektromágneses tekercsek sorozatából áll. Ezek a tekercsek, más néven motortekercsek, felelősek a mágneses mező létrehozásáért, amely kölcsönhatásba lép a rotor mágneses mezőjével. A mágneses mezők kölcsönhatása miatt a rotor diszkrét lépésekben forog. (Athani, 1997)

2.6. Ipar 4.0 és kapcsolódó technológiai

Az Ipar 4.0 napjaink egyik legnépszerűbb hívószava a gazdasági fejlődésnek. De természetesen a társadalom nagy részének sem ismeretlen a kifejezés. Szinte naponta hallani az új fejlesztési ötleteket a témában. Egyre inkább az „okos” kulcs szó használata váltja fel az Ipar 4.0 összefoglaló nevet. Folyamatosan készülnek a tanulmányok az okos otthonokról, okos városokról, okos gyárakról, önvezető járművekről. A tanulmányok egyik központi témája a technológiai fejlődés hatása a munkaerőpiacra.

A világ egy óriási változáson megy keresztül, amely talán az eddig ipari forradalmakhoz képest a legnagyobb. Minden az adatokról, illetve a kapcsolatról, tehát a hálózatba kötött rendszerekről szól. (Geissbauer at al., 2016)

A Roland Berger tanácsadó cég tanulmánya alapján: „az Ipar 4.0 szimbolizálja a negyedik globális ipari forradalmat, amely három technológiai innováció alkalmazásával – automatizálás, Internet of Things, mesterséges intelligencia – alkot úttörő ipari és gazdasági modelleket.” (Berger, 2016)

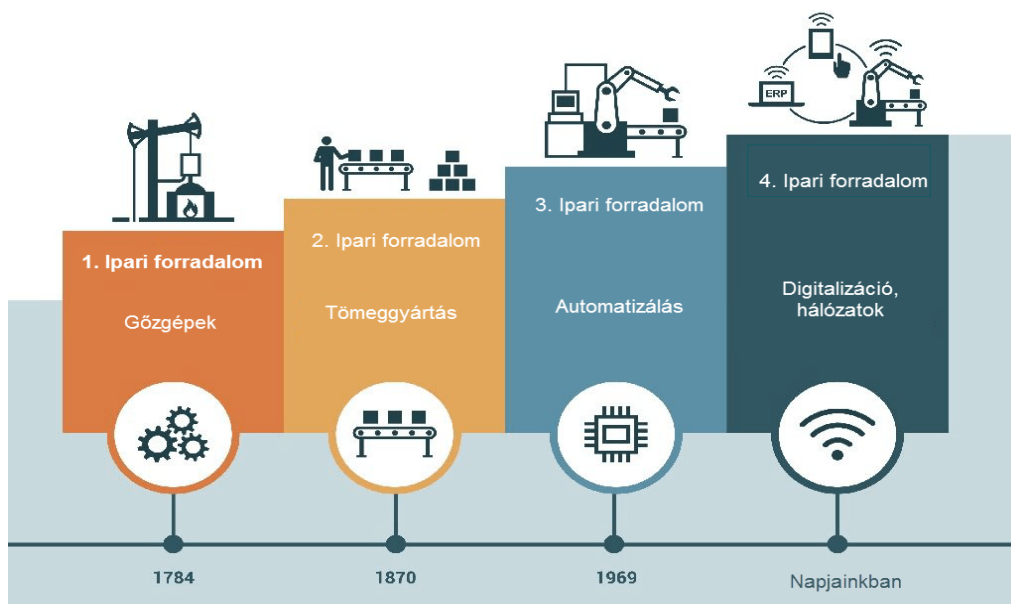
2.6.1. AZ IPARI FORRADALMAK ÉS VÍVMÁNYAIK

A fejlett társadalmak eddig három ipari forradalmat élhettek át teljesen az eddig eltelt évek során, egyet pedig részben, mivel ez most is folyamatosan zajlik. Az első ipari forradalom a lassan 250 éve történt a víz, illetve gőzhajtású gépek megjelenésével. Középpontjában tehát a mechanizáció állt, vagyis megjelent a gépgyártás.

A második ipari forradalom a villamos áram és a futószalagos gyártás alkalmazásával kialakult az igazi tömegtermelés. A harmadik ipari forradalmat pedig a kezdetleges számítógépek általi NC vezérlések okozta automatizáció jelentette.

A negyedik napjainkban is zajló ipari forradalom pedig nem egy eszköz feltalálásával érdemelte ki az „Industry 4.0” címet, hanem a technológia olyan alkalmazásával, hogy a gépek egyetlen közös információs hálózatba kötve, intelligens hálózati rendszerbe integrálódnak. Az Ipar 4.0, mint elnevezés először Németországban jelent meg, ahol adott volt mind a technológia,

mind a gyártás erős jelenléte. Ezt 2011-re datáljuk, lényege pedig a technológiák stratégiai célú felhasználása a hálózati összekapcsolás által. (Tripolszki, 2019)



11. ábra: A négy ipari forradalom legfőbb újításai (Tripolszki, 2019)

2.7. AI alapú gépfelügyeleti rendszerek

A modern világban a gépfelügyeleti rendszerek elterjedtek a különböző iparágakban, és elengedhetetlen technológiát jelentenek ahhoz, hogy a kezelők értékes adatokat és információkat szerezzenek a gépről, például az állapotukról és az életciklusukról. A gépfelügyeleti rendszerekkel a prediktív karbantartás sokkal könnyebbé válik, mivel a gyárak valós időben észlelhetik, elemezhetik és megoldhatják a géppel kapcsolatos problémákat, így a vállalkozások időt és pénzt takarítanak meg. A Statista szerint a globális mesterségesintelligencia szoftverpiac a 2022-es évben 51 milliárd dollárról több mint kétszeresére, 2025-re pedig hihetetlenül nagyra 126 milliárd dollárra fog nőni. (Thormundsson, 2022)

2.7.1. IPARI MESTERSÉGES INTELLIGENCIA

A mesterséges intelligencia (AI – Artificial Intelligence) az emberi intelligencia folyamatok szimulálása gépekkel, különösen számítógépes rendszerekkel. Ezek a folyamatok magukban foglalják a tanulást (az információ megszerzését és az információk felhasználására vonatkozó szabályokat), az érvelést (a szabályokat használva, hogy közelítő vagy határozott

következtetéseket érjenek el) és az önkorrekciót. Az AI speciális alkalmazási területei szakértői rendszerek, beszédfelismerés és gépi látás. Az AI gyenge vagy erős kategóriába sorolható. A gyenge AI egy olyan rendszer, amelyet egy adott feladathoz terveztek és fejlesztenek. Az erős AI, más néven mesterséges általános intelligencia, egy általánosított humán kognitív képességekkel rendelkező AI rendszer. Egy ismeretlen feladat bemutatásával az erős AI rendszer képes emberi beavatkozás nélküli megoldást találni.

Az AI koncepciói és alkalmazásai már hosszú múltra tekintenek vissza. Az elmúlt években az AI iránti érdeklődés megújult az Ipar 4.0 és hasonló mozgalmak összefüggései iránt. Ez azzal a ténnyel magyarázható, hogy a hardverteljesítményben és az erős globális kommunikációs hálózatok elérhetőségében a közelmúltban bekövetkezett fejlesztések számos területen kiterjesztik az alkalmazások körét. Az új irányok, olyan változatos területek, mint az autonóm járművek, ügyfélszolgálat, orvosi diagnosztika és kezelés és pénzügyi tranzakciók. A digitális átalakulás történetében a mesterséges intelligencia szintén kulcsszerepet játszik. (Fritzsche, 2021)

A gépi tanulást és az AI-t a történelem során ördögtől valónak tartották. Sokszor nem álltak rendelkezésre meggyőző bizonyítékok arra vonatkozóan, hogy meggyőzzék az ipart arról, hogy ezek a rendszerek folyamatosan és megbízhatóan működni fognak. Ugyanakkor a gépi tanulási algoritmusok teljesítménye nagymértékben függ a fejlesztő tapasztalataitól és preferenciáitól. Ezért az AI sikere az ipari alkalmazásokban korlátozott. Szisztematikus módszertanként és diszciplínaként működik, hogy megoldásokat kínáljon ipari alkalmazásokhoz, és hídként funkcionáljon, amely összeköti a mesterséges intelligencia akadémiai kutatási eredményeit az ipari szakemberekkel. (Lee et al., 2018)

2.7.2. A MESTERSÉGES INTELLIGENCIA TÍPUSAI

Arend Hintze, a Michigan Állami Egyetem integratív biológiai, számítástudományi és mérnöki asszisztense, az AI-t négyféle kategóriába sorolja: a mai napig létező AI-rendszerektől az érzőrendszerekig, amelyek még nem léteznek. (Rouse, 2020)

- 1. Reaktív gépek:** Erre jó példa a Deep Blue, az IBM sakkprogramja, amely Garry Kasparovot a világbajnok sakkozót, megverte az 1990-es években. A Deep Blue képes azonosítani a sakktáblán lévő darabokat és előrejelzéseket készíteni, de nincs memóriája, és nem tudja a múltbeli tapasztalatokat felhasználni a jövőbeni tájékoztatáshoz. Elemzi a lehetséges lépéseket - sajátját és ellenfelét - és kiválasztja a

legjobb stratégiai lehetőséget. A Deep Blue-t és a Google AlphaGO- t szűk célokra tervezték, és nem könnyen alkalmazható más helyzetre.

2. **Korlátozott memóriával rendelkező gépek:** Ezek az AI rendszerek a múltbeli tapasztalatokat felhasználhatják a jövőbeli döntések tájékoztatására. Az önjáró autók döntéshozatali funkcióinak egy részét ilyen módon tervezték meg. Ezeket a megfigyeléseket nem tárolja véglegesen a rendszer.
3. **Az elmével rendelkező gépek:** „Az elme elmélete”. Ez a pszichológiai kifejezés arra a megértésre utal, hogy másoknak saját meggyőződésük, vágyuk van és szándékai, amelyek befolyásolják a döntéseiket. Ez a fajta AI még nem létezik.
4. **Önismerettel rendelkező gépek:** Ebben a kategóriában azok az AI-rendszerek tartoznak, melyeknek értelmük, öntudatuk van. Azok a gépek, amelyek önismerettel rendelkeznek, megértik a jelenlegi állapotukat, és felhasználhatják az információt arra, hogy másnak érezzék magukat. Ez a fajta AI még nem létezik. (Rouse, 2020)

2.7.3. GÉP ÁLLAPOTFELÜGYELETI RENDSZER

A gyártási folyamatok egyik legkellemetlenebb incidense a gépállás, mind a termelékenység, mind a jövedelmezőség szempontjából. Ha egy gépre oda kell figyelni, az egy teljes ellátási lánc leállítását okozhatja, ami jelentős költséget jelent a vállalkozásoknak mind időben, mind pénzben. Egyszerűen fogalmazva, a gépfelügyeleti rendszer lehetővé teszi egy berendezés folyamatos értékelését, hogy többet megtudjon a teljesítményéről. Ez megkönnyíti a gyári gépek működésében fellépő minták észlelését és az esetleges jövőbeni problémák meghatározását.

A karbantartás minden vállalkozás kulcsfontosságú része. A berendezések nem tarthatnak örökké, és időnként ki kell cserélni vagy javítani kell, hogy a technológiai fejlesztésekhez igazodjanak. Természetesen jobb, ha tisztában van a berendezés állapotával, és a gépfelügyeleti rendszerek kiküszöbölik a találgatások szükségességét, így a tulajdonosok valós idejű betekintést kapnak a gép állapotába, és még azt is lehetővé teszik számukra, hogy előre jelezzék, mikor van szükség karbantartási munkára. Ez azt jelenti, hogy a gyárak jól felkészülhetnek erre az idővesztésre, miközben az állásidőt is minimálisra csökkentik.

2.7.3.1. Állapotfelügyeleti rendszer működése

Egy gépfelügyeleti rendszer hatékony működéséhez, az eszközök és gépek összekapcsolt rendszerének - a dolgok internetének (IoT) - részét kell képeznie. Egy gyárban többféle,

hálózatra csatlakoztatott eszközt szerelnék fel a megfelelő gépekre, azok megfigyelésére. Ezek az érzékelők valós idejű adatokat gyűjtenek megszakítás nélkül, és minden adatot és információt visszaküldenek egy központi számítógépes rendszerbe. Ez a folyamat lehetővé teszi a gyárvezetők számára, hogy megalapozott döntéseket hozzanak a hatékonyság és a termelékenység maximalizálása érdekében. Az érzékelők azt is képesek észlelni, ha a gépi javításokat idő előtt kell elvégezni.

2.7.4. KARBANTARTÁS ÉS MESTERSÉGES INTELLIGENCIA KAPCSOLATA

A karbantartások három kategóriába sorolhatóak: reaktív, tervezett és proaktív. A reaktív karbantartás a gép meghibásodása utáni javítása. A tervezett karbantartás egy berendezés időszakos felmérésének ütemezése annak érdekében, hogy elemezze annak állapotát a problémák észlelése érdekében. Ezeket az értékeléseket az állapottól függetlenül hajtják végre. A proaktív karbantartás során át kell tekinteni az elemzéseket és megérteni a gép kapacitását. Az adatokat a döntések meghozatalához használják fel, de csak azokat a tényezőket veszik figyelembe, amelyek a gép meghibásodását okozhatják, és valójában nem figyelik a gép állapotát, és nem készítenek jelentést. (Portley, 2023)

Ez az oka annak, hogy a mesterséges intelligencia, sikeresen bevezetésre került az iparban. A mesterséges intelligencia újraalkotta a karbantartást, mert most már folyamatosan betekintést nyerhetünk a gép állapotába. A mesterséges intelligencia azt is megjósolja, hogy egy gép mikor igényel figyelmet, ami azt jelenti, hogy a karbantartást csak szükség esetén végzik el, így a költségek alacsonyabbak lesznek. Ezen túlmenően, a mesterséges intelligencia meghatározhatja a gép életciklusát, azt, hogy mikor érdemes javítani, és hogyan tervezheti meg a karbantartási időszakot, hogy a lehető legnagyobb mértékben kiküszöbölje az állásidőt. (Portley, 2023)

A McKinsey & Company jelentése szerint a mesterséges intelligencia alapú prediktív karbantartás akár 25%-kal csökkentheti az ellenőrzési költségeket és 10%-kal az éves karbantartási költségeket, valamint akár 20%-kal növelheti a rendelkezésre állást. (McKinsey, 2017)

A mesterséges intelligencia alapú gépfelügyeleti rendszerek előnyei:

- Kevesebb állásidő,
- Alacsonyabb energiaköltségek,
- Magasabb teljesítmény,
- Csökkentett gyártási költségek,
- Kevesebb hulladék.

2.7.5. AZ ÁLLAPOTFELÜGYELETHEZ HASZNÁLT LEGGYAKORIBB SZENZOROK

Az állapotfelügyeleti rendszerben alkalmazott leggyakoribb szenzorok 6 fő csoportba sorolhatók. A használt szenzor a géptől és a mérendő paramétértől függ:

- A **rezgésérzékelők** mérik a rezgések frekvenciáját, hogy ellenőrizzék a szabálytalanságokat, és előre jelezzék a jövőbeli problémákat. Hasznos motorokhoz, generátorokhoz és turbinákhoz.
- A **nyomásérzékelők** mérik a nyomáskülönbségeket, beleértve a vizet és a levegőt. Hidraulikán használatos.
- A **hőmérséklet-érzékelők** ellenőrzik a berendezés által termelt hőmennyiségét. Számos elektromos készülékhez használják, például légkondicionálóhoz, kemencékhez, vagy hűtőszekrényhez.
- A **páratartalom-érzékelők** hasonlóak a hőmérséklet-érzékelőkhöz, de képesek érzékelni a légkör nedvességtartalmát, és bizonyos esetekben még a levegőben lévő víz pontos mennyiségét is.
- A **gázérzékelők** nyomon követik a különböző gázok szintjét, így biztosítva a biztonságos környezetet. (Portley, 2023)

2.7.6. FESTO AX RENDSZERE

A Festo, a világ vezető automatizálási beszállítója, és elkötelezett a műszaki oktatás iránt is. Felelős családi tulajdonú vállalat és globális szereplő a digitalizációban. Szolgáltatásait a digitalizáció során az intelligens termelés jövőbeni kihívásaihoz igazítja. A vállalat a mesterséges intelligenciára és a gépi tanulásra is támaszkodik.

Az 1950-es években a Festo lett az első olyan vállalat Európában, amely sűrített levegőt használt hajtóközegként az automatizálásban. A vállalat immár több mint 30 000 terméket és rendszermegoldást kínál pneumatikus és elektromos automatizálási technológiához, amelyek a moduláris rendszerek széles választékának köszönhetően, számos különböző gyári és folyamat-automatizálási szegmensben egyedi vevői alkalmazásokhoz szabhatók. Ide tartoznak a pneumatikus és elektromos hajtások, szelepek, szervó vezérlők, mozgásvezérlők, csatlakozástechnikai elemek, kezelési és összeszerelési technológiák, levegő-előkészítő berendezések, szerelvények, vákuumtechnikai elemek, helyzet- és minőségellenőrzők, szenzorok. (Portley, 2023)

Az intelligens termékek, a csatlakoztathatóság, az adatok bányászata és értelmezése, többek között a felhőn keresztül, valamint a vizualizációs irányítópultok már hozzáadott értéket kínálnak az ügyfelek számára. Az olyan termékek, mint az E2M energiahatékonysági modul, az IO-Link kompatibilis alkatrészek, a CPX-IOT átjáró vagy az olyan interfészek, mint az OPC-UA, hozzájárulnak ehhez a folyamathoz. A sikeres és következetes digitalizáció másik alapvető feltétele a mechanikus, elektromos és intelligens összekapcsolhatóság szoftveres megoldásokon keresztül, amelyek lehetővé teszik, hogy minden ügyfél gyorsan és intuitív módon tájékozódjon. (Portley, 2023)



12. ábra: Festo AX

A Festo Automation Experience (AX) egy okos gépfelügyeleti rendszer, amely mesterséges intelligencia és gépi tanulás alkalmazásával maximalizálja a termelékenységet a szenzorok által gyűjtött adatok segítségével. A megoldás használatával a gyárak növelhetik a termelékenységet, csökkenthetik az energiaköltségeiket, és az termelt hulladékot is.

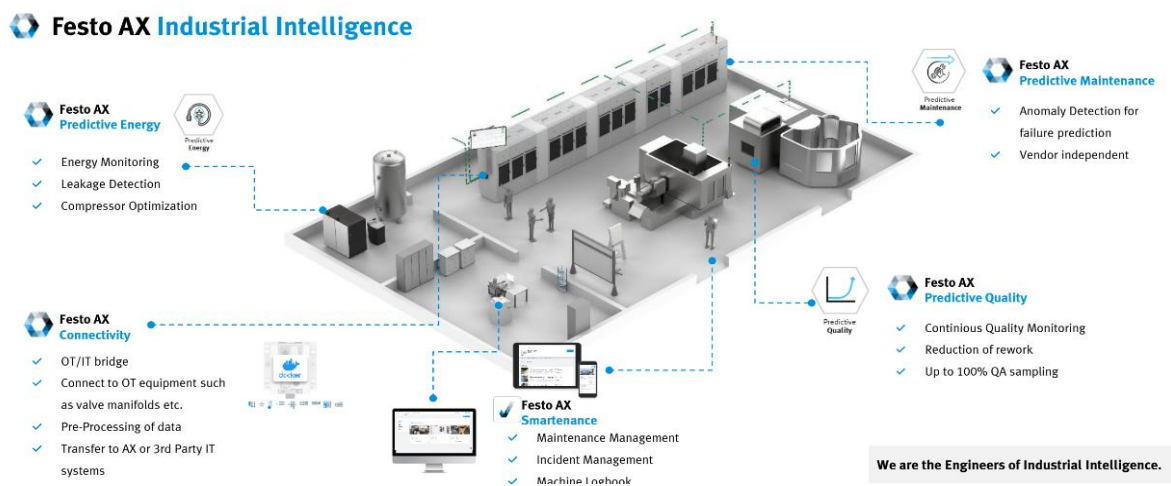
A Festo AX rendszert a következő képességek teszik egyedivé:

- Valós idejű mesterséges intelligencia használat: A Festo AX valós időben elemzi az adatokat. Ez azt jelenti, hogy azonnal, késlekedés nélkül értesül a problémákról. Ha azonnal értesül ezekről az információkról, akkor felkészülhet minden velük kapcsolatos következményre, mielőtt az bekövetkezne.
- Bárhol, bármikor kéznél van: A Festo AX megoldás használata mindenhol támogatott, legyen szó a peremről (a rendszer), a helyi infrastruktúráról (a szerverek) vagy a felhőről.
- Saját magunk rendelkezünk az adatainkkal: A Festo által gyűjtött adatok kizárólag az ügyfél tulajdonát képezik, és kizárólag a megállapodás szerint használhatók fel.

- Rugalmas integráció: Festo AX minden rendszerbe integrálható, mivel tökéletesen együttműködik a különböző gépekben és rendszerekben alkalmazott Festo termékekkel. Az olyan szabványos protokollok használatával, mint az MQTT, ez a megoldás egyszerűen integrálható harmadik féltől származó alkatrészekbe és gépekbe. (Portley, 2023)

Ezek a képességek teszik a lehetővé, hogy a rendszer támogassa a vállalatokat, hogy tisztában legyenek jelenlegi folyamataik állapotával, előre tervezzék karbantartásaikat az adatok elemzése alapján, és energia költségeiket is optimalizálják. A rendszer ezekhez segítséget nyújt a következőképpen:

- Prediktív karbantartás: A rendszer továbbítja az adatokat a Festo AX szoftverbe, és Ön azonnal értesítést kap, ha bármilyen rendellenesség tapasztalható a gép állapotának felügyelete során.
- Prediktív energia: A Festo AX az energiahatékonyság maximalizálásában is segíthet. A szoftver időben figyelmezteti Önt egy, a költségek szempontjából releváns határérték várható átlépéséről.
- Prediktív minőség: A minőségromlás egy másik fontos probléma, amelynek megoldásában a Festo AX képes segíteni. A Festo AX folyamatosan nyomon követi a termék minőségét befolyásoló összetett tényezőket, majd a módosítási javaslatokról jelentést küld a felhasználónak.
- Smartenance: A Smartenance segítségével kezelheti és dokumentálhatja karbantartási feladatait. Az intelligens felhasználókezelés és a csapatmunka hatékonyabbá teszi a karbantartási folyamatokat. Az ösztönös kezelésnek köszönhetően a Smartenance egy egyszerű és költséghatékony szoftver, amivel könnyen digitalizálhatunk. (Portley, 2023)

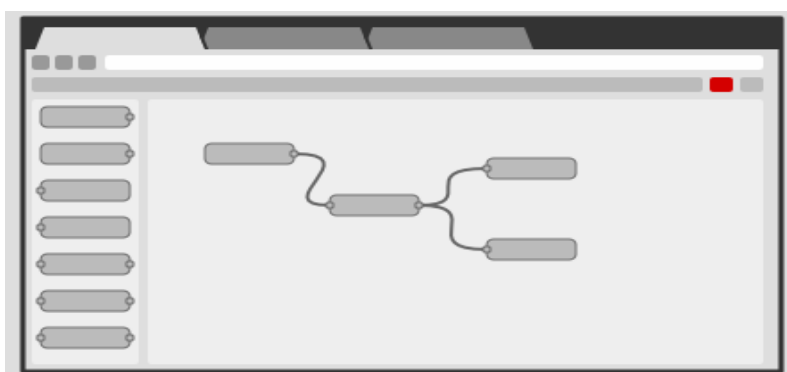


13. ábra: Festo AX előnyei

Az egyik nagy bajor autógyártónál a megmunkáló központok ajtaját nyitó munkahengerrel rendszeresen probléma volt. A sorja rendszeresen szorulást okozott, de változó időtávokban. Volt, amikor 3 hónap után, volt, amikor 9 hónap után kellett cserélni a hengert. De mivel nem volt kiszámítható a használati idő és egy meglehetősen drága alkatrészről volt szó, ezért idő előtt sem szerették volna cserélni. A kiszámíthatatlan csereperiódus pedig azt eredményezte, hogy rendszeresen nem tervezett állásidőt okozott a gépen a cseréje. Az AX telepítése után a rendszer megfigyelte és előre jelezte, hogy lassan csereérett a henger, így a tervezett karbantartáskor tudták cserélni. Ezzel pénzt és erőforrást spóroltak meg.

2.8. Node-RED böngésző alapú grafikus programozás

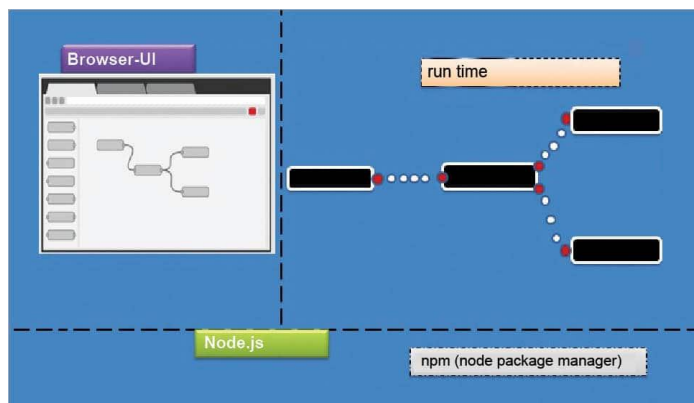
Az IoT technológia számos fejlesztéshez vezetett a vizuális programozás és a vizuális kódolás területén. Ennek eredményeként a már népszerű Node.js-n (szerveroldali Java Scripting platformon) alapuló vizuális programozási/kódoló eszközzé fejlődött, amely főként a tárgyak internete területét célozza meg.



14. ábra: NODE-RED webes mintafelülete (Nodered, 2023)

A Node-RED egy programozási eszköz hardvereszközök, API-k és online szolgáltatások összekapcsolására. Elsősorban a Dolgok Internetére (IoT) tervezett vizuális eszköz, de más alkalmazásokhoz is használható. Nyílt forráskódú, és eredetileg az IBM Emerging Technology által fejlesztete. Az IBM Bluemix IoT-indító alkalmazáscsomag része. A Node-RED külön is telepíthető a Node.js alkalmazás segítségével. Jelenleg a Node-RED a JS Foundation projektje. (Nodered, 2023)

Itt megfigyelhetjük, hogy a böngészőben létrehozott szálak és csomópontok, hogyan alakulnak kóddá a node.js segítségével, ami a háttérben fut. A Node-RED-ben megkülönböztethetünk injekciós csomópontot és függvény csomópontot. Az injekciós csomópontok bemenet nélkül is képesek létrehozni üzenetet, a függvény csomópontok pedig rendelkeznek bemenettel és végrehajtanak rajta változásokat. A Node-RED folyamat úgy működik, hogy üzeneteket ad át a csomópontok között. Az üzenetek egyszerű JavaScript objektumok, amelyek bármilyen tulajdonsággal rendelkezhetnek. (Nodered, 2023)



15. ábra A Node-RED architektúrája (Nodered, 2023)

A Node-RED főbb jellemzői:

- Támogatja a böngésző alapú folyamatszerkesztést.
- Helyben is futtatható Docker támogatás esetén.

A Node-RED számos alkalmazásban használható:

- IoT-hez való csatlakozáshoz.
- Adatbázisokhoz való kötéshez és csatlakozáshoz.
- IoT-adatok tárolására jelen és jövőbeli számításokhoz. (Nodered, 2023)

2.9. Az MQTT protokoll

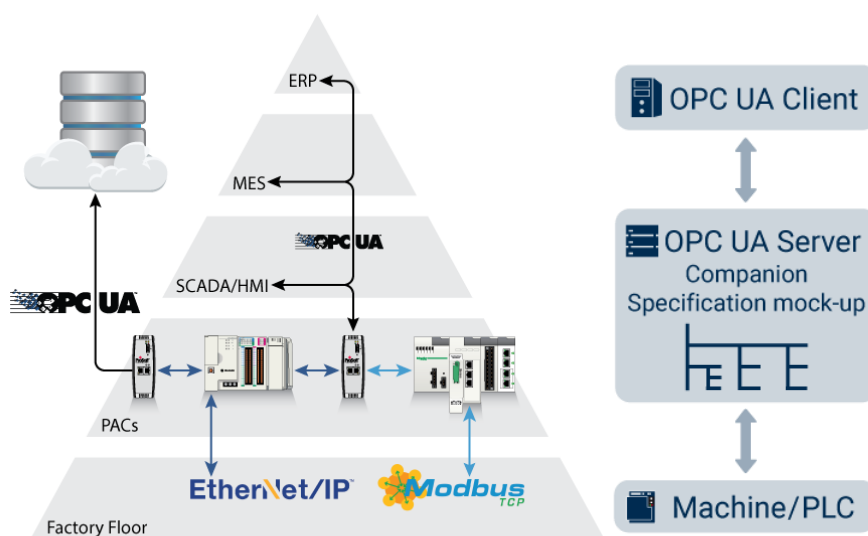
Az MQTT a tárgyak internete (IoT) leggyakrabban használt üzenetküldési protokollja. Az MQTT a Message Queuing Telemetry Transport rövidítése. A protokoll egy olyan szabálykészlet, amely meghatározza, hogy az IoT-eszközök hogyan tehetnek közzé adatokat és fizethetnek elő az interneten keresztül. Az MQTT-t az IoT és az ipari IoT (IIoT) eszközök, például beágyazott eszközök, érzékelők, ipari PLC-k stb. közötti üzenetküldésre és adatcserére használják. A protokoll esemény-vezérelt, és a közzététel/feliratkozás (Pub/Sub) minta használatával köti össze az eszközöket. A küldő (Pub) és a fogadó (Sub) a topic-on keresztül kommunikál, és el

vannak választva egymástól. A köztük lévő kapcsolatot az MQTT bróker bonyolítja le. Az MQTT bróker kiszűri az összes bejövő üzenetet, és helyesen osztja szét azokat. (Hivemq, 2023)

2.10. OPC-UA szabvány

Az egyik legfontosabb kommunikációs szabvány az Ipar 4.0 és az IoT számára. Az IoT-projekt elindításához és futtatásához hozzá kell férnie az IoT-ben található különböző eszközökből, érzékelőkből és gépekből származó adatokhoz, és kommunikálnia kell velük. Szüksége van egy szabványos kommunikációs formára mindegyikben. Itt jön be az OPC UA protokoll. Az OPC-vel az ipari környezetben lévő gépekhez, eszközökhöz és egyéb rendszerekhez való hozzáférés szabványosított, és gyártótól független adatcserét tesz lehetővé.

Ebben az összefüggésben az UA az OPC UA-ban az „Unified Architecture” rövidítése, és a szabvány legújabb specifikációjára utal. Elődjétől abban különbözik, hogy platformfüggetlen, és a COM/DCOM-tól eltér, a tisztán bináris TCP/IP-re vagy a SOAP-ra hasonlít leginkább. (Prosoft, 2023)



16. ábra. OPC-UA szabvány (Prosoft, 2023)

3. MESTERSÉGES INTELLIGENCIA ALAPÚ GÉPFELÜGYELETI RENDSZER FEJLESZTÉSE

A vizsgált szortírozó manipulátor, amelyen a felügyeleti rendszer telepítve lesz, a Festo Kft. váróaulájában található, amely egy preparált szortírozási műveletet mutat be. Lego építőelemeket előre kiválasztott mintázat alapján képes elhelyezni egy 3x1-es mátrixban, melyet a HMI felületen adhatunk meg. 5 darab ilyen elem található a gépben, amely mindegyike különböző színű, és egy szám található rajta egytől ötig. A berendezésben a megfogó egységek kivételével léptető motorokat tartalmaz, amelyek a lineáris és forgó mozgásokért felelősek. Megkülönböztethetünk első és hátsó lineáris tengelyeket, melyek horizontális és vertikális irányban képesek továbbítani a szimulált alkatrészeket. Az első és hátsó tengelyek között egy forgóasztal cseréli az elemeket 90 fokos osztással, így négy alkatrész befogadására képes egyszerre. Található még egy forgóasztal a berendezésben, ami pufferként szolgál az ötödik elem tárolására, és csak a hátsó tengely képes hozzáférni. A Lego alkatrészek azonosítása optikai elven színfelismeréssel történik, de HMI felületen a számokkal azonosítjuk őket, amik szintén megtalálhatóak rajta. A léptető hajtások mindegyikéhez tartozik egy vezérlőegység, illetve a pneumatikus megfogók egy kisebb szelepsziget által vannak működtetve, és végállás szenzorokkal vannak felszerelve. A logikai egységet egy kompakt PLC testesíti meg. Illetve található egy nyomásszenzor is, amely a bejövő sűrített levegő tápnyomásának ellenőrzésére szolgál.



17. ábra. A szortírozó manipulátor

3.1. A válogatási feladat leírása

A berendezést a HMI-n keresztül tudjuk vezérelni. Az inicializálás és az elektromos aktuátorok alaphelyzetbe állítása után a berendezés felveszi a 90 fokos forgóasztal 4 tároló részébe helyezett 4 különböző színű Lego építőelem helyzetét és azonosítja őket a színhez tartozó számmal, ami a PLC programban kerül tárolásra. Az ötödik kimaradt elem a 180 fokos forgóasztal egyik tárolójában helyezkedik el. Pozícióját szenzor érzékeli, de a színét, mely a hozzá társított szám meghatározáshoz szükséges, kizárásos alapon azonosítja. A felmérés és alaphelyzetre állás után választhatunk, hogy melyik három számot, és milyen sorrendben szeretnénk látni a berendezés elején kialakított emelvényen, majd start gomb megnyomásával indítható a szortírozás. A kirakodás után lehetőségünk van azonnal visszatetetni az építőelemeket. Ezután lehet másik mintázatot választani.

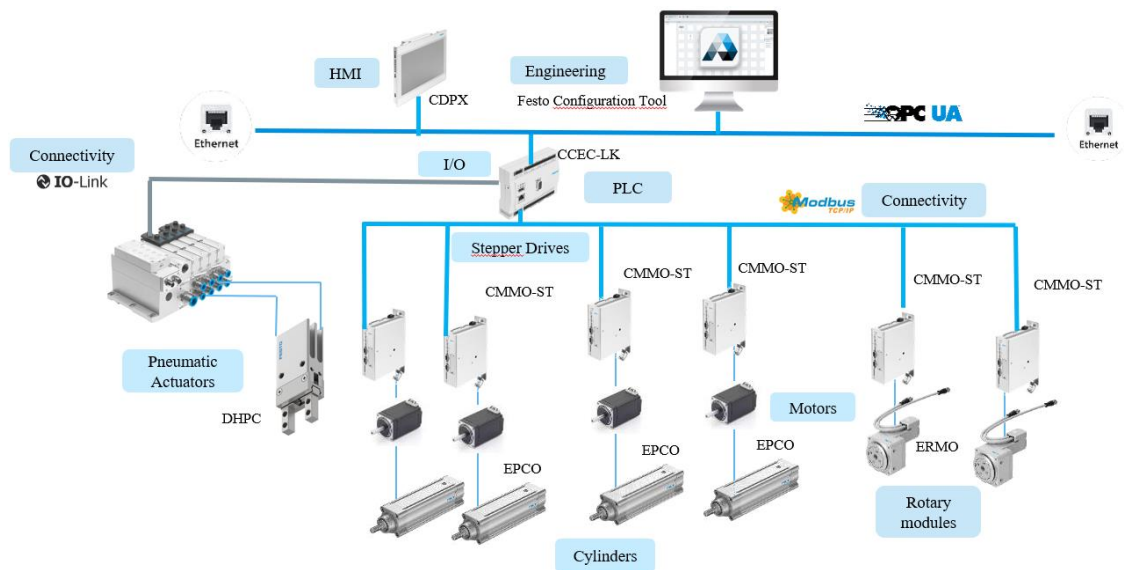


18. ábra A HMI választó és szerviz felülete

3.2. A belső kommunikációs hálózat felépítése

A berendezés belső kommunikációs hálózaton jól bemutatható, hogyan épül fel az ipari berendezések csatlakoztatása, és hogyan valósítják meg az együttműködést az egyes perifériák. A PLC egy egyszerűbb kivitelű kompakt PLC, melynek bemeneteire vannak kötve a végálláskapcsolók és a nyomásszenzor, amely a bejövő levegő tápnyomását hivatott ellenőrizni. A PLC mind a hat léptető hajtás vezérlőjével egy switch-en keresztül kommunikál ethernet, Modbus

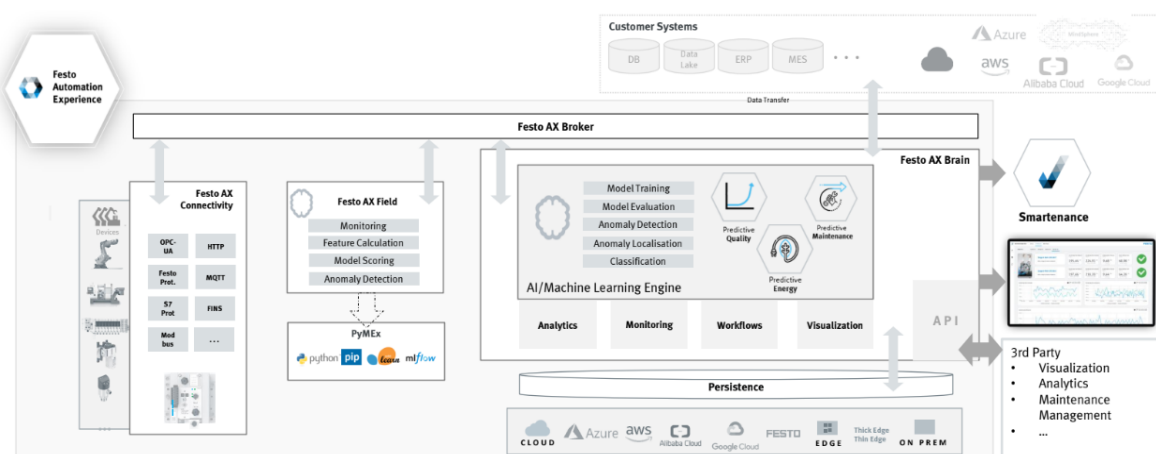
TCP/IP protokollt használva. A szelepszigethez IO-Linken keresztül képes a PLC az utasításokat eljuttatni.



19. ábra A szortírozó manipulátor kommunikációs hálózata

3.3. Az Automation Experience architektúrája és működése

Az AX a gyártási folyamatokból képes adatokat befogadni a különböző eszközöktől a rendszerben, amik különböző típusú adatsomagok és protokollokkal kompatibilis. Többek között optimálisan kommunikál a OPC-UA-val, vagy MQTT-vel, amelyet majd én is használni fogok. De megemlíthetjük a S7 protokollt is, amely a Siemens sajátja, vagy az EtherCAT például Beckhoff, ModbusTCP, HTTP. Itt gyártásban jelenlévő összes működő szenzor jelét képesek vagyunk az AX számára olvasható adattá alakítani.



20. ábra Automaton Experience architektúrája

Az AX Broker, mint kommunikációs csomópont hajtja végre az egyes futó szolgáltatásokat. Az AX Brain hivatott az úgynevezett gépi tanulási folyamatokért. Itt történik a folyamatmodell készítés és értékelés, a hibaészlelés és elemzés, valamint az anomáliák osztályozása. Az itt keletkező adatokat képesek vagyunk az AX rendszeren belül is megjeleníteni, de akár harmadik fél által készített eszközökre is továbbíthatjuk. Az vállalatirányítási rendszerekkel vagy karbantartási szoftverekkel is összekapcsolhatjuk, ha erre van igény, de üzleti elemzőszoftverekkel, mint a PowerBI, vagy MES rendszerekkel is képes kommunikálni. Elmondható róla, hogy minden szinten mindenbe bele lehet illeszteni, hogy felhasználjuk képességeit.

3.4. Egyrendeléses gyártás módosítása folyamatos üzemre

A szortírozó manipulátor eredeti PLC programján módosításokat kellett végrehajtani, hogy a gépi tanulási algoritmus számára egy hosszabb adatsor álljon rendelkezésre a tanulásra, azaz modellkészítésre, és ne kelljen folyamatos emberi beavatkozás a betanítási folyamat alatt. A kirakodási és visszarakodási ciklust kellett állandósítani beavatkozásig vagy hibáig. Az eredeti kód PLC_PRG nevű programján kellett a módosítást végrehajtani. A program módosításához Codesys 3.5 SP7 verziót használtam, mivel ebben fajtában elkerülhetőek voltak a kompatibilitási hibák, mivel ebben a verzióban készült az eredeti program is. Az eredetileg megírt funkcióblokkokat felhasználtam a program írásakor. A program az elején inicializál és alaphelyzetbe állítást végez el hajtásokon, és felveszi a kiindulási pozíciókat, valamint vizsgálja, hogy rendelkezésre áll-e a megfelelő levegőnyomás a működéshez. Ezekkel nem foglalkozom a továbbiakban mert nem én hoztam létre őket. Az előbbi funkciók végrehajtása a fő CASE szerkezet 0, 5, 20, 30,33-as állapotában történnek. Az 10000-es állapot pedig egy hiba kijelzésért felelős része a programnak. A 40-es állapot került újraírásra az új követelményeknek megfelelően, amelyet egy állapotgépbe hoztam létre az átláthatóság kedvéért. Fontos számunkra, hogy ne kelljen mintázatot választani minden egyes ki és visszarakási ciklus végén, hanem a folyamat magától ismétlődjön megállításig.

Bevezetem a state változót és 0-s kezdőértékkel deklarálom. A CASE szerkezet első 0-ás állapotában a feltöltöm a három kiválasztandó lehetőség értéket az 1-es, 2-es, 3-mas elemmel,

```
40:
CASE state OF
0:
    IF aiColor_Request[0]=0 THEN
        aiColor_Request[0]:=1;
        aiColor_Request[1]:=2;
        aiColor_Request[2]:=3;
    END_IF
    fbMapping_Colors_at_Start.bExec := TRUE;
    fbMapping_Colors_at_Start();
    IF fbmapping_colors_at_start.bDone THEN
        state:=20;
    ELSIF fbmapping_colors_at_start.bError THEN
        iselect:=10000;
    END_IF
```

21. ábra PLC program 0. állapot

ha előtte nem került kiválasztásra valamelyik szám. Ezért hasznos mert így elkerülhető a hiba-lehetőség indításnál. Majd elindítom és meghívom a színeket és pozíciókat feltérképező funkcióblokkot. Ha sikeresen lefutott a színérzékelés, akkor tovább lépünk a következő 20-as állapotba, ha nem akkor az 10000-es hiba kijelzési funkció kerül futtatásra.

A 20-as állapotban leállítom a színérzékelő és pozíció meghatározó funkcióblokkot és összeadom az elején automatikusan megadott azonosítókat, hogy ellenőrizhető legyen, hogy ki van-e választva legalább mind a három pozícióra egy elem, amelyre azért lesz szükség, hogyha későbbiekben módosítható legyen és esetlegesen egy hiba jelenséggé értelmezze majd az AX. A színérzékelés az általam írt programban minden kirakodás előtt lefut, így vizsgálva, hogy még az helyükön találhatóak az építőelemek, vagy esetlegesen már nincs is ott mert kiesett valamilyen oknál fogva. Ezután tovább léphetünk a 30-as állapotba.

```
20:
fbMapping_Colors_at_Start.bExec := FALSE;
fbMapping_Colors_at_Start();
iColourSum := aiColor_Request[0] + aiColor_Request[1] + aiColor_Request[2];
state:=state+10;
```

22. ábra PLC program 20. állapot

A 30-as állapot megírása előtt szükségünk van egy időzítőre a visszarakó ciklus ütemezésére és egy élfigyelésre a START gombra a kirakó ciklus indításához. Továbbá a folyamatos üzem érdekében a START gombot átugró változót kellett bevezetni. Ez a bool típusú, "run" nevű változó felelős az indításért és a leállításért is. A START gomb a felfutó élfigyelés miatt megváltoztatja a "run" változó értékét a "run" negált értékére. A PLC első cikluslefutása esetén a "run" változó értéke false, így szükséges a START gomb megnyomása, aminek hatására a "run" értéket vált, true lesz. Ameddig nem nyomjuk meg újból a START gombot, a változó marad igaz állapotban

```

run: BOOL;
figy:r_trig;
END_VAR
=====
fbBackPack_Delay();
figy(clk:=bStart_pack);
IF figy.Q THEN
    run:= NOT run;
END_IF

```

23. ábra. PLC program élfigyelés és időzítő

A 30-as állapotban már a kirakodási funkcióblokk indítása és meghívása történik, de előtte vizsgálom a szín és pozícióérzékelés sikeres lefutását és hogy a START gomb megnyomásra került-e már. Valamint itt került kiértékelésre, hogy ha később módosítunk az alap 1-es, 2-es, 3-mas kombináción, akkor biztosan lett kiválasztva 3 különböző szám. Ha a kirakó

30:

```

IF run AND bColours_Mapped_at_start AND (iColourSum >= 6) THEN
    fbPack_To_Nests.bExec := TRUE;
    fbPack_To_Nests();
    IF fbPack_To_Nests.bError THEN
        iSelect := 10000;
    END_IF
    IF fbPack_To_Nests.bDone THEN
        bStart_pack := FALSE;
    END_IF
ELSE
    fbPack_To_Nests.bExec :=FALSE;
    fbPack_To_Nests();
END_IF
IF fbpack_to_nests.bDone THEN
    fbPack_To_Nests.bExec :=FALSE;
    fbPack_To_Nests();
    state:=50;
ELSIF fbpack_to_nests.bError THEN
    iselect:=10000;
END_IF

```

24. ábra PLC program 30. állapot

funkcióblokk végzett, akkor leállítom és ugorhatunk az 50-es állapotba. Ha nem, mert valamiért hibára futott, akkor 10000-es állapotba kerül, a hiba kiírása miatt.

Az 50-es állapotban a visszarakodási ciklust írtam meg. Itt fogom felhasználni az időzítőt, amit az elején létrehoztam és meghívtam. Leállítom a visszarakó funkcióblokkot, és elindítom az TON típusú bekapcsolásra késleltetett időzítőt öt másodperces időablakkal. Az eredeti visszarakás gomb megnyomásával történt eddig, nekünk pedig ez nem lenne megfelelő a folyamatos gyártási folyamat szimulálására. itt nincs feltétel egyből átléphetünk az 51-es állapotba. Ha idő letelt akkor visszaállítom 0-ra az időzítőt és átléphetünk a 60-as állapotra. Itt elindítom a és meghívom a visszarakodó funkció blokkot. Ha sikeresen lefutott a folyamat akkor léphetünk az utolsó 80-as állapotba, egyéb esetben az 10000-es diagnosztikai részbe. A 80-as állapotban leállítom a visszarakodó funkcióblokkot és 0-s állapotba léptetem a programot. Így elkészült a folyamatos üzemre alkalmas program.

```
50:
    fbNests_to_ERMO90.bExec :=FALSE;
    fbNests_to_ERMO90();
    fbBackPack_Delay(pt:=T#5000MS);
    fbBackPack_Delay(in:=1);
    state:=state+1;
51:
    IF fbBackPack_Delay.Q THEN
        fbBackPack_Delay(in:=0);
        state:=60;
    END_IF
60:
    fbNests_to_ERMO90.bExec := TRUE;
    fbNests_to_ERMO90();
    IF fbNests_to_ERMO90.bDone THEN
        state:=80;
    ELSIF fbNests_to_ERMO90.bError THEN
        iselect:=10000;
    END_IF
80:
    fbNests_to_ERMO90.bExec := FALSE;
    fbNests_to_ERMO90();
    state:=0;
END_CASE
```

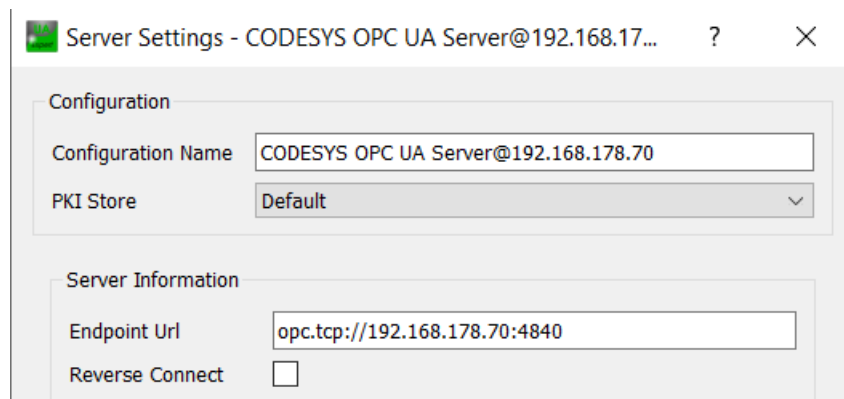
25. ábra PLC program 50.-80. állapotok

Még egy javítást hajtottam végre a első pneumatikus megfogó mozgásáért felelős funkcióblokkon, ahol ugyanis a beépített végállaskapcsolók nem voltak használva a sikeres zárás és nyitás visszaellenőrzésére, de ezt a hibák generálása fejezetben mutatom be.

3.5. OPC-UA szerver és kliens összekötése

Ahhoz, hogy a későbbiekben képesek legyünk a PLC-ben kezelt változók értékeihez hozzáférni és használni, szükségünk van egy OPC-UA szerverre melyet a PLC-n tudunk futtatni néhány beállítást követően. A Codesys programban fül alatt symbol configuration-ben meg kell határoznunk, hogy mely változókat szeretnénk kiküldeni az OPC-UA szerverünkre. Itt azokra a változókra van szükségünk, amik alapján a legjobban megfigyelhető a folyamat, és tudunk következtetni esetleges anomáliákra folyamatos üzem mellett, ugyanis, ha megállna a berendezés, ahhoz nem szükséges egy mesterséges intelligencia, hogy megállapíthassuk, hogy probléma van. Itt én az összes változót kijelöltem, mert így később, majd bármelyiket felhasználtam a AX-re kapcsolás alatt. Ahhoz, hogy a kimenő adatokat elérjük, szükségünk egy úgy nevezett kliensre, ami kiolvassa nekünk. Ehhez az UAExpert nevű programot használtam, mert ez ingyenesen használható és letölthető.

Az UAExpertben csatlakoztattam a PLC OPC-UA szerverét, melyhez szükségem volt a PLC IP címére, amelyet Festo Field Device Tool segítségével könnyedén meghatározhattam. A PLC IP címe 192.168.178.70, amit egyszerűen az opc.tcp:// után kellett megadni így már majdnem elérhető a szerver. Szükség van port megadására is a végső beállításhoz, ami a 4840-es és a Endpoint URL-ként adjuk meg a teljes elérési utat. A sikeres csatlakoztatás után már láthatjuk is az UAExpertben a symbol configuration-ben kijelölt változókat.



26. ábra OPC-UA szerver konfigurálás

3.6. Rendelkezésre álló adatok az OPC-UA-n

Mint az kiderült számomra, ezek a Festo CMMO-ST típusú motorvezérlők pozíció vezérelt módban nem képesek magukról csak a pozíció adatot megosztani a PLC-vel, hiába lehetne látni rengetek más érdekes és számomra releváns adatot a Festo configuration Tool-ban, mint például

a hőmérsékletük, áramfelvételük, és a mozgások paraméterei, a sebesség és gyorsulás adatok. A modernebb CMMT-ST típusok már képesek erre, de sajnos a gép nem ezekkel került felszerelésre. Így a hajtások esetén az aktuális pozíciókat, a pneumatikus megfogók esetén a végál-láskapcsolók igaz-hamis jelét használtam fel, valamint a nyomásszenzor kapcsoltságát. Ezek az adatok is szemléletesek tudnak lenni, ha ráengedünk egy gépi tanulási algoritmust, ugyanis ezek időfüggő változatai már szemléletesek. A megfigyelés alatt az AX rendszer idő függvényében vizsgálja majd a bemenő adatokat a modellkészítésre.

3.7. A Festo AX futtatási környezete

A szortírozó manipulátor mellé egy mini PC-t üzemelt be Festo, amelyre Linux operációs rendszer került telepítésre. Ezen a számítógépen fut a AX minden alkalmazása többek között a Brain, a Field, Iots, valamint a Node-RED csatlakoztatási felület is. Az egyes szolgáltatások Docker konténerben vannak használva, így bármilyen hiba esetén csak az adott szolgáltatást kell újraindítanunk. A konténerizáció hasonlít egy virtualizációra, de ebben az esetben csak az általunk beállított erőforrást és fájlrendszert látja az alkalmazás. Sokszor szokták operációs rendszer szintű virtualizációnak is hívni.

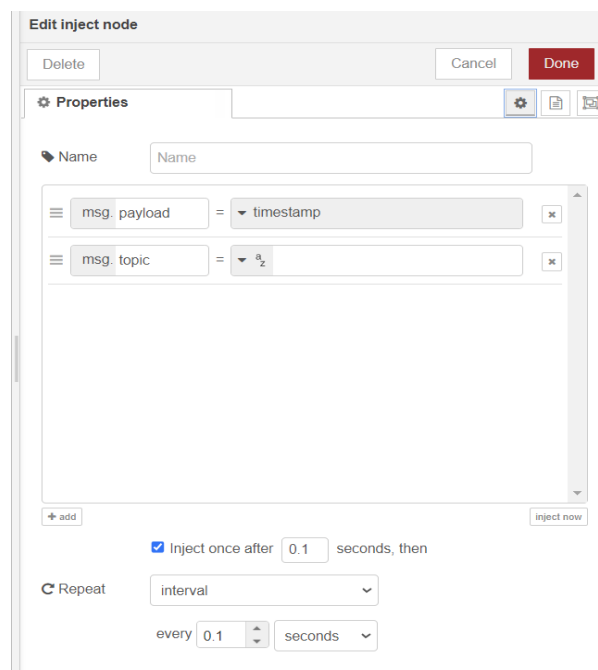
Service	Last known status	Release	
brain	Running	0.0.0+rev17	▶ ■ ↺ 📄
field	Running	0.0.0+rev17	▶ ■ ↺ 📄
iots	Running	0.0.0+rev17	▶ ■ ↺ 📄
mosquitto	Running	0.0.0+rev17	▶ ■ ↺ 📄
nginx	Running	0.0.0+rev17	▶ ■ ↺ 📄
nodered	Running	0.0.0+rev17	▶ ■ ↺ 📄
postgresdb	Running	0.0.0+rev17	▶ ■ ↺ 📄
pymex	Running	0.0.0+rev17	▶ ■ ↺ 📄
springadmin	Running	0.0.0+rev17	▶ ■ ↺ 📄

27. ábra. A PC-n futó alkalmazások

3.8. A Node-RED csatlakoztatási felület

Mint azt korábban említettem a Node-RED egy grafikus programozási környezet, amit én a OPC-UA szerver és az Festo AX közötti kapcsolat megteremtésére használtam MQTT protokoll segítségével. A Linux PC Node-RED alkalmazását egy tetszőleges böngészőn keresztül bejelentkezés nélkül elérhetjük, a 192.168.178.195/nodered címen. Itt az alap fejlesztői környezet fogad bennünket.

Elsőnek a 90 fokos forgóasztal aktuális pozícióját vittem fel, mint továbbítandó adat az AX felé. Ehhez szükségem volt egy injekciós csomópontra, még hozzá egy timestamp vagyis idő-



28. ábra. Injekció beállítása

bélyegzőre, mint kimenő adat. Erre azért van szükség, hogy Node-RED triggerelve legyen az OPC-UA szerverről való olvasásra. Az msg.payload ennek a csomópontnak a tulajdonsága. Valamint a frissítési periódust is beállítottam, hogy milyen időközönként küldjön ki újra időbélyegzőt. Itt 0,1 másodperc elegendő, mert nincs szó olyan gyors mozgásokról. Kezdetben 0,5 másodperccel kezdtem, de ott nem volt megfelelő az AX felületen a folyamatosság.

A következő lépés a OPC-UA item csomópont hozzáadása és összekötése az injekciós csomóponttal. Itt kell beállítanunk a kívánt változó OPC-UA azonosítóját, amely egy elég hosszú és bonyolult karakterlánc. Ami jelen esetben így néz ki: [ns=2;s=|var|CECC-LK.Application.PLC_PRG.ERMO_90_act_degree_int]. Ezt az UAExpert programban a változót megke- resve olvashatjuk ki. A változó típusát is meg kell adnunk ebben is segítségünkre van a UAEx- pert. Megfigyelésem alapján ez mindig string-ként tárolódik, akármilyenként is volt a PLC programban deklarálni.

29. ábra. OPC-UA Item beállítása

Eddig megadtuk már hogy milyen sűrűn és milyen azonosítójú üzenetet szeretnénk olvasni, most arra van szükségünk, hogy ezt honnan tegyünk meg. A következő csomópont, amit hozzá adtam az az OPC-UA Client, ahol a OPC-UA kliens elérhetőséget kellett beállítanunk. Itt a már ismert szerver elérési IP címet és portot kellett használni, `opc.tcp://192.168.178.70:4840`. Ezzel már tudjuk is olvasni az adatot a PLC-ről, amit az akciót read-re beállítottuk. Ezen a ponton a webes felületen egy debug csomópont segítségével ellenőrizhettem is az eddigi folyamatot, ami sikeresen mutatta az aktuális pozíciót. illetve a OPC-UA Client alatt megjelenő active reading is mutatta a sikeres olvasás visszaigazolását.

30. ábra. OPC-UA Client beaállítása

31. ábra. MQTT Client beállítása

Következő lépés az üzenet átalakítása és elnevezése. Itt egy change csomópontot használtam melyben két move szabály használatával átalakítottam az eddigi payload tulajdonságát az üzenetnek egy azonosítható és az AX-ben is értelmezhető adattá, amely az ermo nevet kapta. Az időbélyegzőt ts-re cseréltem, mivel az MQTT-nek szüksége van időadatra, hogy tudja mikor történt a változás. Ezután beállítottam egy téma azonosítót is, amire szintén szüksége van az AX-nek a beolvasáshoz. Ezután egy újabb debug csomópontot használtam és összekötöttem a csere függvényemmel, hogy lássam, hogy sikeresen megtörténtek a változtatások. A debug üzenetben minden információt helyesnek találtam és hibát sem mutatott.

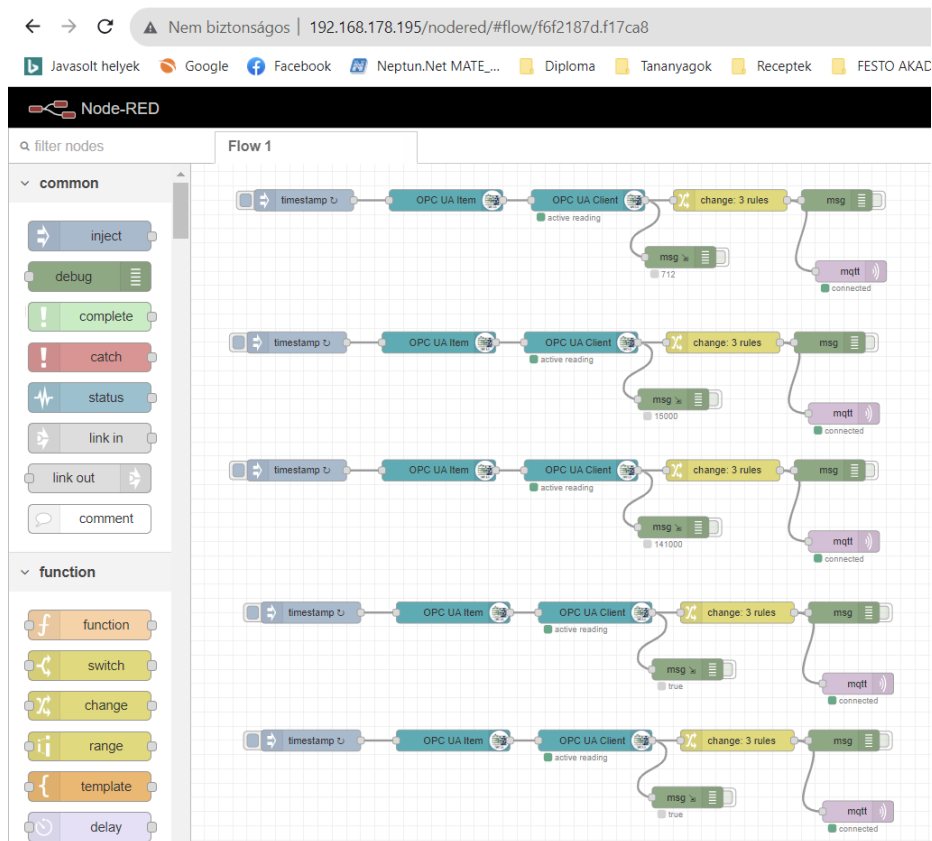
32. ábra. Change csomópont beállítása

A végső csomópont, amit használtam az MQTT kliens, amivel az Node-RED beépítetten rendelkezik. Itt beállítottam a MQTT bróker-t, ami az üzenet küldést végzi az egyik az AX felé. Az AX bróker a mosquitto jelen esetben, amit láthattunk a futó alkalmazások között is a xx ábrán. Az alapértelmezett port nem TSL kapcsolat esetén a 1883-as, így ezt is rögzítettem. Itt

még szükség van a tulajdonságok fül alatt hitsítő adatok megadására is, hogy az AX beengedje az adatokat. Egy felhasználó és jelszó megadást követően el is készült az első Node-RED program. Ezt a folyamatot megismételve az összes kívánt változóra el kell végezni. A rendelkezésre álló változókból a következőket választottam ki, mint számunkra és az AX számára is informatív adatokat:

- Aktuális pozíció:
 - Első tengely vertikális és horizontális hajtásoktól,
 - Hátsó tengely vertikális és horizontális hajtásoktól,
- Aktuális szöghelyzet:
 - 90 fokos körasztaltól,
 - 180 fokos körasztaltól,
- Igaz-Hamis állapotok:
 - Nyomásszenzortól,
 - Színérzékelés és pozíció meghatározás funkció bloktól,
 - A pneumatikus megfogók végállaskapcsolóitól.

A fenti leírás alapján elkészítettem az összes változóra a Node-RED felületen a kívánt programokat. Ezeknek egy részét láthatjuk a 33. ábrán, ahol az is megfigyelhető, hogy a OPC-UA



33. ábra. Node-RED programrészlet

kliensből kilépő debug csomópont mutatja az aktuális pozíciókat, valamint a MQTT kliens csatlakoztatva van a AX rendszerhez. Így már rendelkezésre állnak adatok a mesterséges intelligencia számára, tehát folytathattam a fejlesztést az AX környezetben.

3.9. A szortírozó manipulátor adatainak beolvasása a Festo AX rendszerbe.

Az AX rendszer kezelőfelülete szintén böngésző alapú, így könnyen csatlakozhattam a 192.168.178.195 IP cím megadása után. Ez felhasználói menedzsment támogatott rendszer így, egy saját fiók elkészítése után már használhattam is a kezelőfelületet. A felület felhasználóbarát és logikus felépítésű.

A saját Lego elnevezésű projekt létrehozása után a MQTT klienssel kiküldött üzenetet kellett beolvasnom. Itt az infrastruktúra, eszközmenedzserben egy eszköz hozzáadáshoz szükség van a change csomópont által beállított, payload után létre hozott névre, és a topic tulajdonságra. Ezek az ermo, és az ermo/szog, a fenti Node-RED minta alapján. Ermo, mert a hajtás Festo ezt az elnevezést használja, illetve szog, mert szögelfordulásról beszélünk körasztal révén. Szükséges továbbá megadni az adat típusát is, ami esetünkben numeric, azaz számérték. Az információ típusánál lehetőségünk van megadni, hogy mire szeretnénk használni a rendszert. Én analízálni szeretném majd ezeket az adatokat, így erre állítottam. Ezt elvégezve az összes a Node-RED-ben is használt változóra elvégeztem, így már az AX is beolvassa az PLC-ről érkező adatokat. A 33. ábrán a felvitt változókat láthatjuk, valamint az elérési utat is, még az MQTT kliensnél állítottam be.

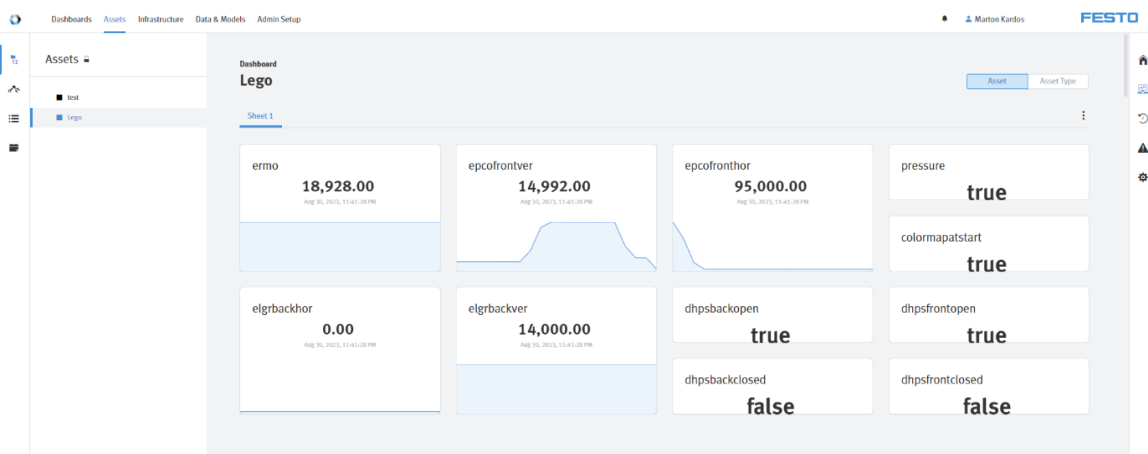
Protocol	MQTT						
IP address	tcp://mosquitto:1883						
Authentication	Password						

Device Features							
						Import	Export
							Delete selected
☐	Name	Unit	Topic	Information type	Data type	Recording	Tags
☐	colormapstart		colormap/done	Analysis	String	On	⋮
☐	dhpsbackclosed		dhps/poz	Analysis	String	On	⋮
☐	dhpsbackopen		dhps/poz	Analysis	String	On	⋮
☐	dhpsfrontclosed		dhps/poz	Analysis	String	On	⋮
☐	dhpsfrontopen		dhps/poz	Analysis	String	On	⋮
☐	elgrbackhor		elgr/poz	Analysis	Numeric	On	⋮
☐	elgrbackver		elgr/poz	Analysis	Numeric	On	⋮
☐	epcofronthor		epco/poz	Analysis	Numeric	On	⋮
☐	epcofrontver		epco/poz	Analysis	Numeric	On	⋮
☐	ermo		ermo/szog	Analysis	Numeric	On	⋮

34. ábra. bemenő adatok az AX rendszerben

3.10. Az AX irányítópult létrehozása és beállítása

Az AX-ben beépített vezérlőpult áll rendelkezésre a megfigyelni kívánt adatok megjelenítésére és monitorozására. Ehhez a Lego projektem dashboard menüjében kell a már felvitt változókat kiválasztani, és engedélyezni, hogy használni szeretném őket. Itt több megjelenítési formát választhattam, de mivel nekem szám adatok állnak rendelkezésemre a pozíciókból és szöveges igaz vagy hamis állítások az égállásokból, ezért grafikon stílusú és szöveges megjelenítőket használtam. Az összes változót kihelyeztem a felületre és minden megjelenített érték a PLC-n levő változók értékeivel egyezett meg ezt ellenőriztem. De ugyebár már a Node-RED felületen is megjelentek, hol szintén lehetőségem volt ezt ellenőrizni. El is indítottam a manipulátort, hogy tesztelhessem, az adatok változását az irányítópulton. Mivel az számértékek elégséges frissítéssel változtak a mozgásoknak megfelelően, így sikeresnek nyilvánítottam az összekötést. A következő részben bemutatom a szortírozó manipulátoron milyen mesterséges anomáliákat lehet létrehozni annak érdekében, hogy a AX-et tesztelni lehessen.



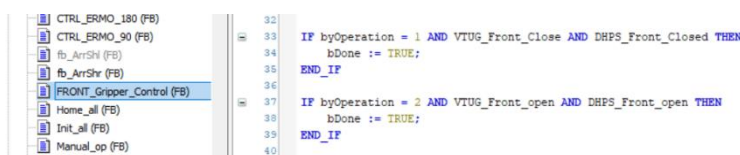
35. ábra. AX vezérlőpult

3.11. Mesterséges anomália generálása a manipulátoron

Az AX mesterséges intelligencián alapuló rendszernek a teszteléséhez hibákat kellett kitalálni és megvalósítani. Ezek a hibák nem lehettek olyanok, amik a rakodási feladatot teljesen megállítják. Mivel ilyenkor a PLC is hibára állna ki és a megállt folyamat észrevétele nem igényel mesterséges intelligenciát. Itt olyan lehetőségeket használtam fel, amik főleg a mozgások sebességére vannak hatással. Ugyanis a valós gyártási folyamatok esetében is ezeket tudjuk extra szenzorozottság nélkül megfigyelni, ha nem rendelkezünk a legmodernebb elektromos hajtásokkal.

Lehetőségek hibagenerálásra:

- Az első tengely pneumatikus megfogójának sebességének állíthatósága miatt beépítettem egy fojtószelepet, amelyet könnyedén tudok állítani akár üzem közben is. Annak érdekében, hogy a PLC ne álljon ki a megfogót vezérlő funkcióblokkban módosítanom kellett a timeout időzítőt nagyobb értékre, amely 5 másodperc lett. Ezt a folyamatot tesztelve azt figyeltem, meg hogy sokszor elejti a Lego építőelemeket a megfogó. Kisebb utána járás után rájöttem, hogy a funkció blokkban a továbblépés feltételénél nincsenek használva a végállaskapcsolók, csak a szelep átváltás van figyelve és ha az váltott a PLC engedélyezi a továbblépést. Ezért a feltételekhez felvettem a végállaskapcsoló nyitott és zárt állapotát visszaigazoló változóit. Így már nem volt probléma a lelassított megfogó esetében sem a rakodás során. Tehát már képes vagyok a megfogó sebességét vezérelni a folyamat megállása nélkül. A módosítást a 35. ábrán láthatjuk.



36. ábra. Végállásfigyelés pneumatikus megfogónál

- A másik lehetőségem az elektromos hajtások dinamikai jellemzőinek állítása volt. Ezt Festo Configuration Tool programban tudtam megtenni. Mivel a vezérlők a PLC-vel egy switch-en keresztül kommunikálnak, a közös hálózatra csatlakozva bármelyik vezérlőt elérhetem és változtathatom beállításait. Az első tengely vertikális és horizontális hajtásain csökkentettem a sebességet 50 százalékkal, valamint a 90 fokos körasztal sebességet csökkentettem 30%. A beállítások után visszaadtam az irányítást a PLC-nek, így tesztelhettem is a beállításokat, amik sikeresen meg is valósultak.

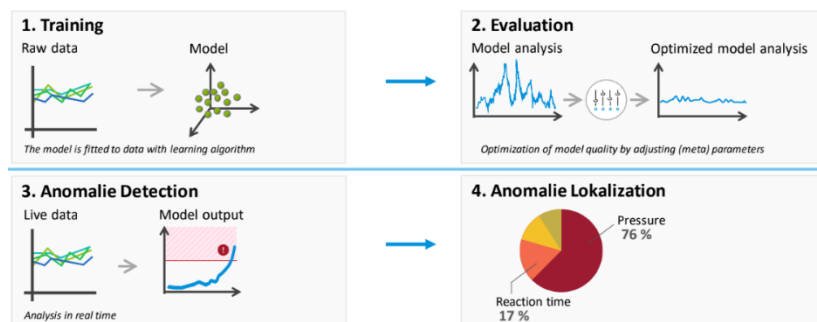
3.12. Sikertelen modellgenerálás és anomália érzékelés

A modell generálást sajnos nem sikerült végrehajtani mivel a számomra biztosított rendszeren valamilyen oknál fogva nem engedte a Festo AX rendszere az adatsor rögzítését és így nem volt lehetőség modellt generáltatni vele. Sajnos a problémát a Festo sem tudta orvosolni a szakdolgozat készítési időszakban, ezért a várt eredményeket mutatom be, és a védésen prezentálom az eredményeket, amennyiben addigra sikerült orvosolni a problémát.

Az AX esetében is, mint minden mesterséges intelligencia alapú algoritmusnál szükséges, hogy feltanítsuk a rendszert, hogy mit tekinthet alapállapotnak, mihez képest kell vizsgálnia a

későbbiekben a változásokat. Tehát a normál üzemi körülmények között, mesterséges hibák generálása nélkül kell járatnunk a berendezést legalább néhány órát. A Festo tapasztalatai alapján a rövid időintervallum nem elegendő megfelelő feltanításhoz.

Itt a nyers adatokat mutatunk az AX-nek és utána a felvett adatsorokból modellt generáltunk vele. A modellgenerálás után szükséges finomításokat elvégezni és optimalizálni a modellünket, mert lehettek olyan adatok a folyamat során, ami már anomáliára utalhat, ezért ezektől megszabadítjuk, hogy ne legyen része az adatmodellünknek. Régen határértéket állítottak be a szakemberek, és ha ezen átlépett valamelyik adat, akkor megvizsgálták. Jelen esetben ezeket a szinteket magától állítja be a AX és módosítja dinamikusan annak megfelelően, amit észlel. Az anomália jelenségeket képes osztályozni és meghatározni, hogy mik a fő veszélyforrások. Képes prediktíven előre jelezni meghibásodásokat, például úgy, hogy a modelljétől eltérő áramfelvételt érzékelve jelez, hogy valamilyen probléma található rendszerben, amit érdemes felülvizsgálni. Ezt természetesen egy ember is észrevenné, ha folyamatosan a hajtás áramfelvételét elemezné, de erre nincs erőforrása egyetlen cégnek sem.



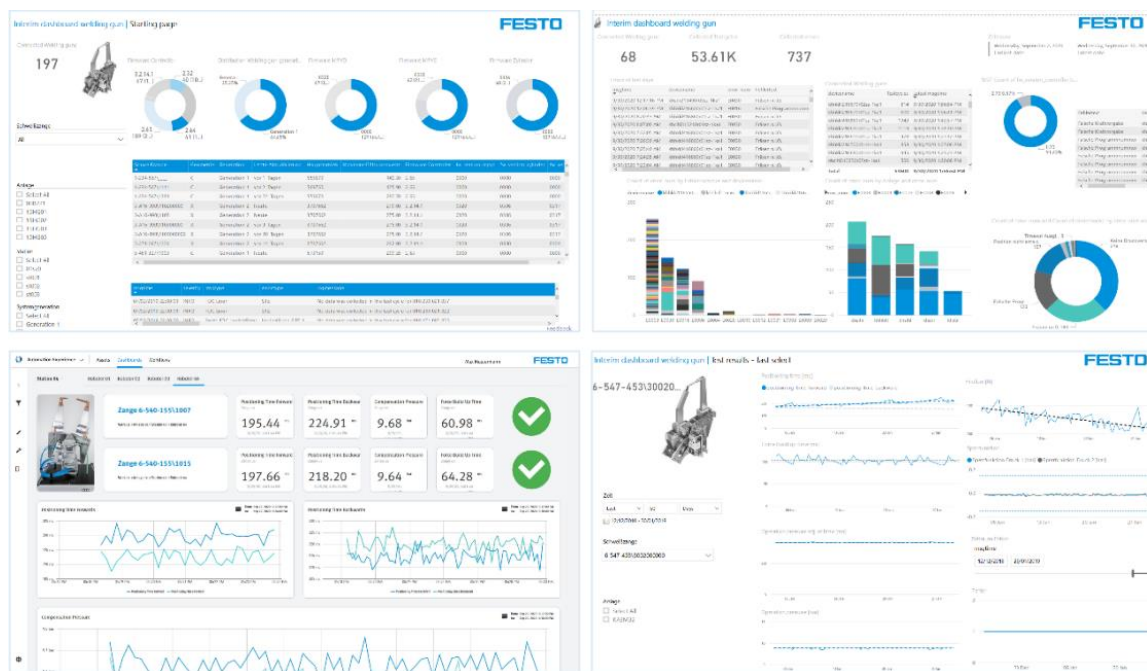
37. ábra. AX működésének folyamatát

3.13. Várt eredmények és konklúzió

Mivel a Festo AX rendszerrel nem volt lehetőségem tesztelni a hibákra adott reakciókat a szakdolgozat elkészítési határidejéig, ezért a várt eredményeket mutatom be. A fő elv az anomália generálásra a sebességek befolyásolása volt, ami a valós gyártási folyamatok esetén is előforduló probléma. Ezt előidézheti például levegőszivárgás a pneumatikus elemek esetén, amit valamilyen tömítési hiba okozhat az energialáncban, vagy valamilyen cső megtörése, ami fojtásként funkcionál. A munkahengerek fojtása valós gyártási körülmények között is bevett szokás a sebesség szabályozásra, ahol egy kisebb elállítás már nagy befolyással lehet a ciklusidőre. Ezeket a hibákat az AX képes lenne kimutatni, és figyelmeztetni a felhasználót, akár egy

email formájában is. Megvalósítható az is, hogy a használt karbantartási rendszerben feladatot generál a karbantartás részére, hogy célszerű a felülvizsgálat az adott helyen. Az elektromos hajtások sebesség csökkentésére szintén hasonló reakciót vertam volna a rendszertől. A berendezésben használt motorvezérlők már javarészt kifutott hajtásai a Festo-nak, így sok olyan funkciót nem támogatnak az adat továbbítására, mint az újabbak. Ezek például a leadott nyomtaték, sebesség és gyorsulás, hőmérséklet, stb. Pedig ezek a fizikai mennyiségek nagyon hasznosak lennének a felügyelet szempontjából. Ugyanis az elektromos berendezések konstrukciónál fogva tudják kompenzálni egy ideig a hibás működést. Nekünk ebben az időszakban, ha információt biztosít az AX rendszer a jelenlegi helyzetről és figyelmeztet, hogy probléma léphet fel, akkor tervezetten van lehetőség az alkatrész cseréjére, ami időt és pénzt spórol meg a nem várt állásidő elmaradása miatt.

Összeségében tehát elmondható, hogy a Festo AX mesterséges intelligencián alapuló rendszere egy hasznos jövőbe mutató technológiai algoritmus, ami pénzt spórolhat meg a felhasználó számára, amennyiben jól szenzorozott folyamatról van szó. Illetve adaptálva karbantartás menedzsment rendszerbe megkönnyítheti a hibák feltérképezését és a javítások tervezését, az emberi erőforrást megspórolva. Véleményem szerint egy ilyen rendszernek helye van az ipari termelésben. Valamint ez a technológia még csak most kezd beszivárogni a gyárakba. Később számíthatunk rá, hogy alapvető eszköze lesz a menedzsment a költségek csökkentésében és a gyártástervezési folyamatok optimalizálásában.



38. ábra AX kiértékelő felület

3.14. Gazdasági számítás

Az AX rendszer licenz díja egy gyártósorra megközelítőleg 20 000 EUR-ba kerül egy évre. Egy ember megtanítása a használatára és a gyártósor feltanítása 1 hónapot vesz igénybe. Így a 12 hónapból 11 hónapot tud a rendszer hasznosan működni az egy éves időszakban. Innentől kezdve 3 műszakos munkarenddel számolva minden műszakban helyettesíteni tud egy embert, akinek a feladata a soron keletkező hibák feltérképezése. Viszont így fordítható másra is a munkaerő. Az AX rendszer tapasztalatok alapján 20 %-kal tudja csökkenteni a nem tervezett állásidőt, ami egy nagyobb gyártóüzem esetében akár óránként 2 000 EUR megtakarítás is lehet. Valamint a berendezés energiamonitorozása is hasznos, mert ebből tudunk következtetni a befektetett költségek egyik fontos paraméterére. Az extra szenzorok felhelyezése körülbelül 10 000 EUR. A mintaszámítás alapján elmondható, hogy az Festo AX rendszere egy kifizető befektetés és számos nem kimutatható előnnyel is rendelkezik a 196 000 EUR-os megtakarításon kívül.

1. táblázat. Számítási összegzés

	Költségek	Nyereségek
Festo AX liszensz díja:	20 000 EUR	
3 ember munkabére 11 hónapra:		66 000 EUR
Szenzorika:	10 000 EUR	
Elmaradt nem tervezett állásidő évente 80 óra		160 000 EUR
Összesen:	30 000 EUR	226 000 EUR
Megtakarított összeg:	196 000 EUR	

4. ÖSSZEFOGLALÁS

Szakedolgozatom fő célja egy modern ipar 4.0-ba illeszkedő technológia, a mesterséges intelligencia alapú gépfelügyeleti rendszer megismerése és fejlesztése volt. Ehhez egy preparált szortírozó manipulátort használtam eszközként, valamint a Festo AX szoftverét.

Első lépésként feltérképeztem a kommunikációs hálózatot. Majd megvizsgáltam a berendezést, hogy milyen eszközök állnak rendelkezésre a kísérlet elvégzéséhez. Ezek alapján terveztem meg a további feladatokat. Az AX rendszernek a teszteléséhez bemenő adatokat kellett biztosítanom a manipulátorról. Így a fő vezérlő egység a PLC által megosztható változókat egy OPC-UA szerver futtatásával kívülről is láthatóvá tettem. Majd az UAExpert programmal rácsatlakoztam erre a szerverre és kiolvastam a kísérlet szempontjából fontos változókat. Ezek a korábban vártnál kevesebb információt hordoztak az elektromos hajtások miatt, mivel ezek régebbi modellek voltak. Így a korábban várt sebesség, gyorsulás, áramfelvétel, nyomatékleadás helyett pozíció adatokat tudtak csak közölni. A változók kinyerése után a Node-RED csatlakoztatási felület segítségével felprogramoztam és összekötöttem az AX rendszerrel az OPC-UA szervert. Ehhez MQTT protokollt használtam a kommunikációra.

Az AX rendszerhez hozzáadtam a kívánt változókat és kialakítottam a megfigyelésre alkalmas kezelőfelületet. A felvitt elemek a 90 fokos körasztal, az első tengely horizontális és vertikális tengely pozíciói, valamint az pneumatikus megfogók nyitott és zárt állapot visszajelző végállás kapcsolók voltak. A helyes adatgyűjtéshez, szükség volt a szortírozó manipulátor folyamatos üzemére, amit az eredeti PLC program nem tett levetővé csak rendszeres emberi beavatkozással. Ezért a programon módosítást hajtottam végre, hogy több órányi szünetmentes adatsor álljon rendelkezésre. Az anomália érzékeléshez és az AX reakcióra bíráshoz szükséges volt olyan mesterséges hibák alkalmazása a gépen, amik nem okoznak leállást, de a valóságban valamilyen helytelen működési állapotra utalnak. Az elektromos hajtások esetében a rekordtáblát módosítva, a pneumatikus megfogó esetében folytatást alkalmazva állítottam át a sebességet. A lassabb sebesség jellemzőt tesztelve lettem figyelmes arra, hogy a megfogók végálláskapcsolói nincsenek használva a PLC programban, így hibás rakodást eredményeztek. Ezt optimalizáltam, ami megoldotta a problémát.

Ezután következett volna az adatok rögzítése és a modell alkotás az AX rendszerben, ami nem valósult meg, mert a Festo sem tudta orvosolni a problémát a dolgozat elkészítési határidejéig, de amennyiben ez a későbbiekben lehetőségem lesz, elvégzem a kísérletet.

Összeségében megállapítható, hogy a mesterséges intelligencia alapú gépfelügyeleti rendszernek komoly jelentősége van a mai korszerű gyártási rendszereknél. Könnyen, gyorsan adaptálható, mivel számos ipari kommunikációs protokollt képes kezelni. Alkalmas számos olyan folyamat elvégzésére, amit eddig akár több szakképzett ember végzett, így erőforrást szabadít fel, és a precizitásával túl is mutat bizonyos tekintetben az emberi képességeken. Pénzt és energiát spórol a gyártási folyamatokon a potenciális meghibásodások előre jelzésével, ami a nem tervezett karbantartást csökkenti.

5. SUMMARY

The main goal of my thesis was to learn and develop a modern technology for industry 4.0, the artificial intelligence-based machine monitoring system. For this purpose, I used a preprepared sorting manipulator as a tool and the Festo AX software.

As a first step I mapped the communication network. Then I examined the setup to see what tools were available to perform the experiment. On this basis I planned the further tasks. To test the AX system, I had to provide input data from the manipulator. Thus, I made the main control unit visible externally by running an OPC-UA server to the PLC. I then connected to this server using the UAExpert program and read out the variables relevant to the experiment. These carried less information than previously expected due to the electric drives being older models. So instead of the previously expected speed, acceleration, current consumption, torque output, they could only give position data. After extracting the variables, I programmed and connected the OPC-UA server to the AX system using the Node-RED connection interface. For this I used MQTT protocol for communication.

I added the desired variables to the AX system and created a sub-interface for monitoring. The elements added were the 90 degree rotary table, the horizontal and vertical axis positions of the front axis, and the pneumatic grippers open and closed state feedback limit switches. For correct data acquisition, continuous operation of the sorting manipulator-reader was required, which the original PLC program did not allow except with regular human intervention. Therefore, I made modifications to the program to make several hours of uninterrupted data available. To detect the anomaly and make the AX react, it was necessary to apply artificial faults on the machine that do not cause a shutdown, but in reality indicate some kind of incorrect operating condition. I adjusted the speed by modifying the record table in the case of electric drives and by applying a flow in the case of pneumatic grippers. When testing the slower speed feature, I noticed that the limit switches on the grippers were not used in the PLC program, resulting in incorrect loading. I optimized this, which solved the problem.

This was to be followed by data capture and model building in the AX system, which was not done because Festo could not rectify the problem by the time of writing this thesis, but if I have the opportunity to do so in the future I will carry out the experiment.

Overall, it can be concluded that AI-based machine control systems are of great importance in today's advanced manufacturing systems. It can be easily and quickly adapted, as it can handle a wide range of industrial communication protocols. It is capable of performing many

processes that were previously performed by several skilled humans, freeing up resources and, in some respects, surpassing human capabilities through its precision. It saves money and energy in manufacturing processes by anticipating potential failures, reducing unplanned maintenance.

6. IRODALOMJEGYZÉK

Anyagmozgatás és raktározás. Letöltés dátuma: 2023.10. 4. Forrás:

<http://vili.pmmf.hu/portal/documents/19217/19796/anyagmozg.pdf>.

Athani V. V. (1997): Stepper Motors: Fundamentals Applications and Design. New Delhi: New Age International

Bánki P., Lovas A. (2012): Szenzorika és anyagai. Budapest : Typotex Kiadó

Berger R. (2018): Hotel Industry 4.0. Bécs: Roland Berger GmbH

Dr. Kodácsy J., Dr. Pintér J. (2011): Szerszámgépek és gyártórendszerek. Budapest: Széchenyi István Egyetem

Dr. Németh I. (2013): Anyagmozgató berendezések. Budapest: Budapesti Műszaki és Gazdaságtudományi Egyetem

Fritzsche A. és Gölzer Ph. (2021): Industrial Applications of Artificial Intelligence. 54th CIRP Conference on Manufacturing Systems. Patras: Elsevier B.V.

Geissbauer R., Vedso J. és Schrauf S. (2016): Industry 4.0: Building the Digital Enterprise. PwC Global Industry 4.0 Survey. Letöltés dátuma: 2023.10.27. Forrás: <https://www.pwc.com/gx/en/industries/industries-4.0/landing-page/industry-4.0-building-your-digital-enterprise-april-2016.pdf>

International standard (2021): ISO 8373:2021 Robotics - Vocabulary. Geneva: Reference number ISO 8373:2021(E)

Korbuly B. (2020): Mi az AI, vagyis a mesterséges intelligencia, és mióta létezik? Letöltés dátuma: 2023. 10 10. Forrás: <https://www.crane.hu/mi-az-ai-vagyis-a-mesterseges-intelligencia-es-miota-letezik>.

Krámli Gy. (2008): Szenzorika. Budapest: Festo Didactic

Kulcsár B., Pápai F. (2012): Az anyagmozgatás irányítás- és automatizálástechnikája. Budapest: Typotex Kiadó

Lee J. et al. (2018): Industrial Artificial Intelligence for industry 4.0 - based manufacturing systems. Manufacturing Letters. Cincinnati: University of Cincinnati

Lee K. (2016): Artificial intelligence, automation, and the economy. Washington: The White House Blog

Mayerné S. E. (2022): Ipari robottechnika. Gödöllő : Magyar Agrár- és Élettudományi Egyetem

McKinsey & Company (2017): Smartening up with Artificial Intelligence (AI) - What's in it for Germany and its Industrial Sector. Kiadó: Digital McKinsey

Online forrás: Nodered.org Letöltés dátuma: 2023.10.30. Forrás: <https://nodered.org/docs/getting-started/local>

Online forrás (2017): Robotok és manipulátorok a rugalmas gyártórendszerekben. uni-miskolc.hu. Letöltés dátuma : 2023.10.31. forrás: http://www.uni-miskolc.hu/~ggy-tmazs/tantargyak/gyartasi%20foly/robotEloadas_2014.pdf

Online forrás: ROHM. Letöltés dátuma: 2023. 10. 30. Forrás: <https://www.rohm.com/electronics-basics/sensor/color-sensor>.

Online forrás: What is Fieldbus? The RealPars Team. Letöltés dátuma: 2023. 10 27. Forrás: <https://realpars.com/fieldbus/>

Portley J. (2023): AI-based Machine Monitoring Systems. Letöltés dátuma: 2023. 10.04. Forrás: <https://knowhow.distrelec.com/automation/ai-based-machine-monitoring-systems/>

Online forrás: OPC-UA gateway to the IIOT, Prosoft Technology. Letöltés dátuma: 2023.10.27 Forrás: <https://www.prosoft-technology.com/Landing-Pages/OPC-UA>

Rostan M. (2014): Industrial Ethernet Technologies: Overview and Comparison. Nuremberg : EtherCAT Technology Group

Rouse M. (2020): A Machine Learning and AI Guide for Enterprises in the enterprise. Letöltés dátuma: 2023. 10 04. Forrás: <https://www.techtarget.com/searchenterpriseai/definition/AI-Artificial-Intelligence>

Siciliano B., Khatib O. (2016): Handbook of Robotics. Stanford: Springer International Publishing

Thormundsson B. (2022): Artificial intelligence software market revenue worldwide 2018-2025. Statista.com, Letöltés dátuma: 2023. 10 27. Forrás: <https://www.statista.com/statistics/607716/worldwide-artificial-intelligence-market-revenues/>

Tripolszki G. (2017): A 4. Ipari forradalom. Letöltés dátuma: 2023. 10 31. Forrás: <https://blog.eplm.hu/4-ipari-forradalom/>

Wilson S. J. (2004): Sensor Technology Handbook. Oxford : Newnes Letöltés dátuma: 2023.10.27. Forrás: https://doc.lagout.org/science/0_Computer%20Science/8_Electronics%20%26%20Robotics/Sensor%20Technology%20Handbook%20-%20By%20J.%20Wilson.pdf

MATE Szervezeti és Működési Szabályzat

III. Hallgatói Követelményrendszer

III.1. Tanulmányi és Vizsgaszabályzat

**6.13. sz. függeléke: A MATE egységes szakdolgozat /
diplomadolgozat / záródolgozat / portfólió készítési útmutatója**

4.1. sz. melléklete: Konzulensi nyilatkozat

NYILATKOZAT

Kardos Márton (név) (hallgató Neptun azonosítója: MSJ3B2)
konzulenseként nyilatkozom arról, hogy a
záródolgozatot/szakdolgozatot/diplomadolgozatot/portfóliót¹ áttekintettem, a hallgatót az
irodalmi források korrekt kezelésének követelményeiről, jogi és etikai szabályairól
tájékoztattam.

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

A dolgozat állam- vagy szolgálati titkot tartalmaz: igen nem³

Kelt: 2023. év 11. hó 09. nap



belső konzulens

¹ A megfelelő dolgozattípus meghagyása mellett a többi típus törlendő.

² A megfelelő aláhúzendő.

³ A megfelelő aláhúzendő.

MATE Szervezeti és Működési Szabályzat

III. Hallgatói Követelményrendszer

III.1. Tanulmányi és Vizsgaszabályzat

**6.13. sz. függelék: A MATE egységes szakdolgozat /
diplomadolgozat / záródolgozat / portfólió készítési útmutatója**

4.2. sz. melléklete: Nyilatkozat a záródolgozat/szakdolgozat/diplomadolgozat/portfólió nyilvános hozzáféréséről és eredetiségéről

NYILATKOZAT

**a záródolgozat/szakdolgozat/diplomadolgozat/portfólió¹ nyilvános hozzáféréséről és
eredetiségéről**

A hallgató neve:

Kardos Márton

A Hallgató Neptun kódja:

MSS3BZ

A dolgozat címe:

Heterogén intelligencia alapú gépfelismerési
rendszer fejlesztése

A megjelenés éve:

2023

A konzulens intézetének neve:

MATE Műszaki Intézet

A konzulens tanszékének a neve:

Mechatronika Tanszék

Kijelentem, hogy az általam benyújtott záródolgozat/szakdolgozat/diplomadolgozat/portfólió² 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 szellemi tulajdon-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: 2023 év 11. hó 05. nap

Kardos Márton

Hallgató aláírása

¹ A megfelelő dolgozattípus meghagyása mellett a többi típus törlendő.

² A megfelelő dolgozattípus meghagyása mellett a többi típus törlendő.

