

# SZAKDOLGOZAT

Michelisz János György

2023

Magyar Agrár- és Élettudományi Egyetem  
Élelmiszertudományi és Technológiai Intézet  
Fizika-Automatika Tanszék

Robotok alkalmazása az élelmiszeripar-  
ban

Michelisz János György

Budapest

2023

# **TARTALOMJEGYZÉK**

1. Bevezetés
2. Munka Célja
3. Irodalmi áttekintés
  1. Ipar 4.0 bemutatása
  2. Ipaer 4.0 előnyei
  3. Az ipar4.0 újszerűtechnológiája
  4. Ipar 4.0 élelmiszeriparban
  5. Robotika
  6. Robotika az élelmiszeriparban
4. FANUC robot programozása
  1. Koordináta rendszerek
  2. Program létrehozása
  3. Robotkart mozgatása
  4. Regiszterek
  5. Kimenetek és bementek. (Output, Input)
  6. Utasítások
5. Palletázó robot programozása
6. Összesítés:
7. Irodalmi hivatkozás

# **1. BEVEZETÉS**

A Robotika az élelmiszeriparban egy fontos téma, amivel érdemes foglalkozni, mert a robotok képesek az ember munkáját helyettesíteni olyan helyeken, ahol automatizálással a gyártás volumene nőne, továbbá a számos előnye van az emberilánc által végzett munkával szemben. A robotok természetesen nem veszik el az embertől a munkát, bár bizonyos munkakörök megszűnhetnek, de más munkák jönnek létre helyette amihez az emberi beavatkozás szükséges. A robotok segítségével az ember képesek más irányba fókuszálni, fontosabb munkákat elvégezni.

Témámnak a Robotika az élelmiszeriparban választottam. A téma azért fogott meg, mert az egyetemi tanulmányim során olyan tárgyak tetszettek a legjobban, amelyek a műveletgyártási témával foglalkoznak. Mászóval, a mérnöki oldalával. Ilyen tárgyam volt például az ipari mérés technika, az élelmiszeripari műveletek, továbbá az irányítástechnika. Ezek a tárgyak nagyon érdekeltek, felkeltették a figyelmemet, értettem őket és szerettem rájuk készülni. Ezeken felül megerősítette a témaválasztásomat az is, hogy egy gyárlátogatás keretein belül betekintést nyerhettem a robotikáról és annak potenciáljáról. Ez a téma nagyon passzolt az elképzelésemhez, miszerint az élelmiszeripart inkább a műszaki oldaláról szeretném megközelíteni, úgyhogy erre esett a választásom.

A dolgozatom során az irodalmi áttekintésben be fogom mutatni, hogy az ipar 4.0-nak milyen hatásai voltak az élelmiszeripariparra, ezen kívül az automatizálással is szeretnék részletesen foglalkozni a szakdolgozat keretein belül. Ezután rátérek arra, hogy bemutassam, hogy hogyan építették be, hogyan tudják felhasználni az automatizálást és a robotikát élelmiszeriparban és az agrár területeken.

Robotika részletes bemutatása során, le fogom írni a robotika történelmét, milyen fajta robotokat ismerünk, mi különbség a robotika és az automatizálás között, milyen iparágakban használják fel legnagyobb volumenben. Pár példán keresztül bemutatom, hogy a robotokat hogyan használják fel az élelmiszeriparban.

A szakdolgozatom gerincét képező része egy általam kitalált feladatot elvégző robotnak a bemutatása lesz. A feladata a robotnak az lesz, hogy műanyag hengereket kell egyesével a számukra kialakított tarolójukba helyeznie, szóval egy palettázó robotot szeretnék programozni.

## **2. A MUNKA CÉLJA**

Munkám fő célja egy működőképes palettázó robotnak a megvalósítása lesz, amit a folyamat részletes leírása mellett képek vagy video formátumban is igyekszem bemutatni. Ennek megvalósításával szeretném mélyíteni tudásomat robotikában és a programozásban.

Ezt a robotot a gyakorlati helyem segítségével szeretném megvalósítani, ugyanis ott lesz lehetőségem a megvalósításhoz szükséges eszközöket és programokat használni.

Ezenkívül, szeretnék mélyebb elméleti tudást és ismeretet szerezni az élelmiszeripar műszaki részéről, a robotokról, az ipar 4.0-ról és az automatizálásról. Milyen mostani lehetőségek vannak az élelmiszeriparban robotizáció szempontjából. Milyen további fejlesztésekre van szükség

### **3. IRODALMI ÁTTEKINTÉS**

#### **1. Ipar 4.0 általános bemutatása**

Az Ipar 4.0 az egy új ipari forradalom, amelynek korát éljük mi is. Mint a többi ipari forradalom ez is pont olyannyira meghatározza az életünket, mint az előző három is. A kifejezést egy német tudósokból álló csoport alkotta meg és mutatta be. (Koczka, 2021)

„Az Ipar 4.0 alatt az automatizált, optimalizált és teljesen integrált folyamatot értjük. A szenzoroknak, munkadaraboknak, gépeknek és az informatikai rendszereknek a horizontális és vertikális kapcsolatát eredményezi a teljes ellátási-lánc mentén.” (Oláh et al, 2019, p.4)

A továbbiak megértéséhez elengedhetetlen néhány fogalom tisztázása. Az első ilyen fogalom a big data, amelynek feladata a szenzorok és szoftverek által folyamatosan gyűjtött hatalmas mennyiségű adatokat optimálisan rögzítse, mentse, megossza, védje az adathalászoktól, szűrje és lekérhetővé tegye. A big data fontos része, hogy az adatok mennyire változatosak és helyszükséglete. (Benedek, 2020) A következő ilyen tisztázandó fogalom a mesterséges intelligencia (MI) fogalma. Ez a számítástechnika egyik ága, amely során a gépek algoritmusok vagy parancskódok segítségével képesek olyan feladatok elvégzésére, amire emberi intelligencia nem képes. Az MI képes tervezni, problémákat megoldani, döntéseket hozni, de akár még tanulásra is képes. A gépek közötti kommunikáció (M2M) azt jelenti, hogy két vagy több gép, vagy robot adatokat közvetít egymásnak, emberi segítség nélkül. A kommunikáció megoldható vezetékes kapcsolatokon keresztül (PLC) vagy vezeték nélküli kapcsolatokon. Utóbbi esetén valamilyen hálózaton keresztül történik az adatok folyása (IoT). Az IoT kapcsolat, angolul Internet of Things, jelentése: dolgok az interneten. Olyan intelligens gépek ezek, amelyek közös hálózaton keresztül képesek kommunikálni egymással. Ez a vezeték nélküli kapcsolat sokkal egyszerűbbé tette a gépek közötti kommunikációt. (Koczka, 2021)

Összegezve, az ipar 4.0 segítségével már képesek lehetnek a robotok és a gépek intelligens döntések meghozatalára, hiszen az adatokat már nem csak gyűjtik, hanem információkat is kinyernek belőle és így képesek belőle döntéseket meghozni (AI). Továbbá, a gépek már egymással is képesek kommunikálni, amellyel folyamatos információ folyást biztosítanak, kiváltva az ember beavatkozásának a lehetőségét és szükségességét (M2M).

Ezen felül az ipar 4.0 által jóval szorosabb kapcsolatban van az információ az automatizálással, így maga a gyártási technológia is megváltozik. (Koczka, 2021.).

## **2. Az ipar 4.0 előnyei**

Adódik a kérdés, hogy a vállalatoknak milyen előnyei származhatnak az ipar 4.0 alkalmazásával. Az egyik legnagyobb előny, amiért a vállalatoknak érdemes ipar 4.0-át használni, hogy magasabb termelékenységet tudnak vele elérni, miközben a költségeiket is tudják vele csökkenteni. Egy teljesen automatizált és intelligens termékgyártási sor jóval gyorsabb sebességgel képes előállítani a termékeket, miközben kevesebb emberi munkaerőt, és optimálisabb anyag és energiafelhasználást biztosít. Ez azért is pozitív, mert a megspórolt költségek és emberi munkaerő által a vállalat jobban allokálhatja az erőforrásait más területekre, ezáltal fejlesztve mind a dolgozóinak a képességét, mind a saját termelékenységét és előretörési lehetőségeit. (Koczka, 2021))

Az ipar 4.0 egy másik nem elhanyagolható tulajdonsága, hogy az elektronikai eszközök folyamatos fejlődésének köszönhetően, a gyártók képesek lehetnek agilis gyártási folyamatokat létrehozni. „Az agilis módszertan olyan gyakorlatot, munkaszervezést jelent, amely során a szoftverfejlesztés vagy egyéb projekt agilis értékek mentén iterációkban zajlik.” (Prohuman consulting, 2022) Az agilis módszertan segítségével folyamatos a kapcsolat a vállalat ügyfeleivel, hamarabb kap a vállalat visszajelzést a termékkel kapcsolatban és így nagyobb eséllyel képes fejleszteni azt, változtatni rajta. Azaz a rugalmas gyártósorok elérhetővé válnak, és így a vállalat képes alkalmazkodni a beszállítók és ügyfelek váltakozó igényeihez. (Koczka, 2021)

## **3. Az ipar 4.0 újszerű technológiája**

Az ipar 4.0-hoz tartozó technológiákat három nagy csoportra oszthatjuk fel. Az első ilyen csoport a felhőalapú rendszerek. Ez alatt azt értjük, hogy nagy mennyiségű és valós idejű adatok érhetők el számunkra (Big data). A második csoport a Kiber-fizikai rendszerek csoportja, amiben a termékek kommunikálni tudnak a gyártás folyamán. Ezt a beépített szenzorok teszik lehetővé, ezáltal segítve azt, hogy további információhoz juthassunk magáról a termékről. A harmadik, végül a legutolsó az intelligens gyár csoportja, ami „az ember és a gép közötti intelligens hálózatba kapcsolódást jelent olcsó automatizálással és valós idejű

adatokkal.” (Oláh et al, 2019, p.6) Ezen területek külön-külön behozatalával és alkalmazásával hatalmas hatékonyságot és termelékenységet lehet elérni, de ezek együttes használata nagy mértékben segítené elő a fejlődést a gazdaság összes területén. (Oláh et al, 2019.)

#### **4. Az ipar 4.0 az élelmiszeriparban**

Az élelmiszeriparban fontos, hogy a nyersanyagokat egészen az eredetükig vissza lehessen követni, és ennek érdekében már számos nyomon követési rendszert fejlesztettek ki és alkalmaznak napi szinten. A cél az, hogy ezek a metódusok gyorsabbak, átláthatóbbak, nyomon követhetőek legyenek és, hogy versenyképesek maradjanak. Az ipar 4.0-ból származó erőforrásokat olyan területekre kell csoportosítani, ahol még nem, vagy csak nagyon korlátozottan használhatnák ilyesfajta megoldásokat.

Az élelmiszeriparban ezeket leginkább az üzemi és gyártási folyamatokban lehet alkalmazni, mint például a folyamatok szabályozása, az üzem higiéniaja működtetése, maga az élelmiszerbiztonság, valamint a logisztikával kapcsolatos feladatok. Az ipar 4.0-ból származó lehetőségek az élelmiszeriparban való alkalmazását és terjedését nehezíti, hogy az ágazat különösen egyedi és heterogén a többi iparággal szemben, ezen felül a különböző termékek is más-más innovációs megoldásokat igényelnek. Ebből kifolyólag új szervezeti formák jelentek meg amelyek segítik, hogy az ipar 4.0-ból származó lehetőségeket integrálják mind a termelésben, mind a feldolgozásban. Ugyanakkor az élelmiszeriparra nagy hatással van a globalizáció is, ami segítheti a technológia implementálását, és így az ipar 4.0 pozitív tulajdonságai képesek kifejteni hatásukat az iparban. Ilyen hatás például a költségcsökkentés, a munkaerő hatékonyság, a termelékenység fokozása, a fogyasztói és beszállítói igényeknek való megfelelés erősítése, és az élelmiszerbiztonság és minőség hatékonyabb szabályozása. (Debrenti, 2020.)

#### **5. A robotika:**

A robotok kialakulása a manipulátorokhoz vezethető vissza. A manipulátorok olyan berendezések, amelyek folyamatosan ugyanazokat a mozgássorozattokat ismétlik, vagy egy emberi kéz irányítja őket. A manipulátorok esetén a program már bele van építve a gépbe és ennek a programnak a megváltoztatását csak nagyon igényes és bonyolult átalakítással lehetne megvalósítani. Ezért szükségessé vált ezen berendezéseknek a tovább fejlesztése, így megalakultak a robotok.(Kovács, 2015)



A robotok olyan gépek, amelyeket általában egy feladatsor megoldására programoznak, de lehetőség van arra is, ha egy más programot töltünk fel a robotba, ezáltal másik feladatot is el tud végezni. Más szóval a robot egy olyan gép, amelynek több funkciója is lehet és ezáltal, hogy mire szeretnék felhasználni, annak csak az emberi képzelet szab határt. (Kovács, 2015)

Ahhoz, hogy a robotokat beépítsük és alkalmazhassuk egy rendszerben, szükséges volt, hogy a robotokkal közölt információk gyorsabban legyenek feldolgozva, mert annak egymás után kell megoldania a feladatait. Ezért Rodney A. Brooks egy új információfeldolgozás architektúrát ajánlott, amely a viselkedésalapú robotika az alapja lett. Ennek az új információfeldolgozásnak a felépítése kis részben hierarchikus. Az összes alrendszer szuverénággént működik és nincs központi tervezés, mindegyik alrendszer csak a saját feladatát kezeli. A folyamat maga az érzékelés, modellezés, tervezés, tervvégrehajtás és mozgásvezérlés lineáris sorozata.

Az érzékeléstől a mozgásvezérlésig tartó információfeldolgozást a blokkok nem sorosan, hanem a modulok párhuzamosan, egyidejűleg végzik el a feladataikat. Az alrendszerek saját érzékelőkkel és beavatkozókkel rendelkeznek, miközben kapcsolatot tartanak a környezetükkel. A kapcsolattartás azt jelenti, hogy más alrendszernek impulzív vagy gátló üzeneteket küldenek, amivel képesek kapcsolatot fenntartani. (Kovács, 2015)

A robotokat sok szempont alapján lehet csoportosítani. Lehet az alapján, hogy melyik generációba tartozik, vagy külső megjelenés szerint (robot, mobil, robotkar), de intelligenciaszint vagy felhasználási terület alapján is. (Kovács, 2015)

A robotokat történelmi előre haladásukkal generációba sorolták. Míg eleinte az 1960-as években az 1. generáció csak tárgyak mozgatására voltak képesek, addigra a 70-es években a 2. generáció robotjai már képesek lettek a környezetük vizsgálatára is, így feladataikat eszerint kellett megváltoztatni. Ma már az ipar 4.0 lehetőségei által a 3. generációs robotokat már képesek a mesterséges intelligenciával ellátni, így saját döntéseket tudnak hozni a vizsgált környezetük alapján vagy a Big Data adataiból. (Kovács, 2015)

A külső megjelenés alapján is képesek vagyunk megkülönböztetni a robotok fajtáit. Vannak a mobil robotok melyek lehetnek androidok, animatok, vagy önvezetésű járművek. Vannak a statikus robotok, amelyek csoportjába tartoznak a háztartási- vagy ipari robotok és a robotkarok. Végül az utolsó csoport a különleges robotok és a nanorobotok csoportja. (Kovács, 2015)

A robotokat a széles felhasználási lehetőségei miatt már számos területen alkalmazzák. Vanak már háztartásban megjelenő robotok, mint például a robot porszívó. Használják gyógyászati vagy orvosi célokra is, amely során akár gyógyszergyártásban vagy műtéti eljárásokban is részt vesznek. Katonai feladatok elvégzésére is alkalmasak a robotok, például sokszor a felderítésekre is robotgépeket küldenek. Ám legnagyobb részben a robotokat az iparban használják, azon belül is a nehézipar és az autóipar a legkiemelkedőbb. (Kovács, 2015)

A robot törvényeket Isaac Asimov fektette le, három alapszabályt meghozva.

Az első, hogy „a robot nem árthat az embernek, és nem nézheti tétlenül, ha az embert veszély fenyegeti”. A második az, hogy „a robot engedelmeskedni tartozik az emberek parancsainak, kivéve, ha ezek a parancsok az első törvénybe ütköznek”. Végül pedig „a robot köteles megvédeni magát mindaddig, amíg ez nem ütközik az első vagy a második törvénybe.” (Kovács, 2015)

A robotok felépítésük szerint két fő részre oszthatóak. Az egyik a manipulátor, aminek a feladata az, hogy a hozzá tartozó végeffektort helyzetbe, orientációba hozza. A másik a magát a robotot irányító számítógép, ami tartalmazza a vezérlőberendezést és a vezérlési algoritmusokat. (Kovács, 2015)

Effektor az, amit a robot „kezeiként” is felfoghatunk, amivel a robotok képesek egy céltárgyat megfogni, vagy egy célszerszámmal valami feladatot elvégezni rajta.

Annak függvényében, hogy milyen energiafelhasználással fogják meg a céltárgyat, a robotokat megkülönböztethetjük vákuumos, pneumatikus (legelterjedtebb), hidraulikus, és szervomotoros fajtái szerint. (Kovács, 2015)

Ahhoz, hogy a robotok képesek legyenek mozgásra, számos lehetőség van. Először is meg kell nézni, hogy a robot milyen hajtással rendelkezik. Ez lehet, mint az effektornál, elektromos, hidraulikus vagy pneumatikus. A robotok minden ízületéhez külön hajtás tartozik és az ízületek mennyisége határozza meg egy robotnak a szabadsági fokát. „A szabadságfok azt mutatja, hogy hány különböző irányba képes a robot mozogni. Minimálisan 3 az elvárható, azonban a gyakorlatban 5, 6 és 7 szabadságfokkal találkozhatunk a leggyakrabban”. (Kovács, 2015)

## **6. Robotika az élelmiszeriparban**

Az élelmiszeriparban az élelmiszergyártó és -feldolgozó gyárak igyekeznek ma már költség-hatékonyabb automatizálási megoldásokat alkalmazni a nagyobb termelési volumen érdekében. A kézi munkaerőt igyekeznek, automatizált vagy esetleg robotizált munkafolyamatokra cserélni. Gyakori példa a csomagolás vagy a raklapozás robotizált megoldása. (Iqbal, 2017)

De a robotokat igyekeznek további munkakörökben is alkalmazni, mint az automatizált minőségellenőrzés. Melyet mai modern látórendszerek, a nagyobb felbontású (HD) kamerán keresztül, a robotok képesek a hibák azonosítására, detektálására, amit fel lehet használni a zöldségek és gyümölcsök érettségének vizsgálatára. (Iqbal, 2017)

A gyártók termelékenysége akár 25% kel nőtt robotika használatával, de a különböző élelmiszeripari ágazatokban eltérhet. Bár ezt sok minden befolyásolja az automatizálás szintje, a telepített robotok száma és a vevői igények változása miatti termékváltoztatás. Emiatt bizonyos követelménykét fektet le az élelmiszeripar robotizálással kapcsolatban. Az első követelmény aminek meg kell felelnie élelmiszeriparnak, a termékváltoztatást, de anélkül, hogy a feldolgozási vonalon változtatni kellene, vagy a hardverekkel való bajlódás. (Iqbal, 2017)

A második a higiénia. Az élelmiszer-biztonság fontos kérdés, és előírják, hogy az élelmiszereket és italokat feldolgozásuk során az ember ne érintse meg, hogy elkerülhető legyen a kórokozók és baktériumok átvitele. Az ilyen szigorú követelményekhez az élelmiszeriparban elengedhetetlen a robotok, látórendszerek és végberendezések vagy megfogók higiénikus tervezése. Az élelmiszer-kezeléshez használt robotok megfogóit ipari mosószerrel és túlnyomásos forró vízzel mossák le. (Iqbal, 2017)

Még fontos tényező a termelékenység és a biztonság. Ezen beruházások megtérülése a volumen növekedésében tudjuk mérni és a kiadások csökkenésében. Amik többnyire alátámasztják a robotizálás fontosságát. De így további veszélynek tesszük ki az ottani dolgozókat, így fontos a balesetmegelőzési oktatások és ezen veszélyes területek elkerítése. (Iqbal, 2017)

## **4. FANUC robot programozása**

A szakdolgozatom fő témája egy robotkarnak egy olyan program írása és futtatása, amivel képes vagyok a robottal palettázó műveletre. A szakmai gyakorlat helyemen erre laboratóriumi szintű megvalósításra volt lehetőségem, de az elméleti háttere ugyan az, csak kis részben tér el az ipari szinttől. A művelet egy Fanuc LR 200 Mate iD nevezetű robot karral végeztem el. A robotkar alaptulajdonságai tartozik, hogy az ipari robotokhoz képest kisméretű, 6 tengelyel rendelkező robot, melynek tengelyei teljesen különböző elforgathatósági szögekkel és elforgathatósági sebességekkel rendelkeznek, amelynek a mérete és maximális kinyúlása megegyezik az emberi karéval(717mm). Max teherbírása 7 kg. Tápja 200-230V 50/60 hz 1 állású váltó áram esetében. Ezenfelül IP67 védelemmel rendelkezik.

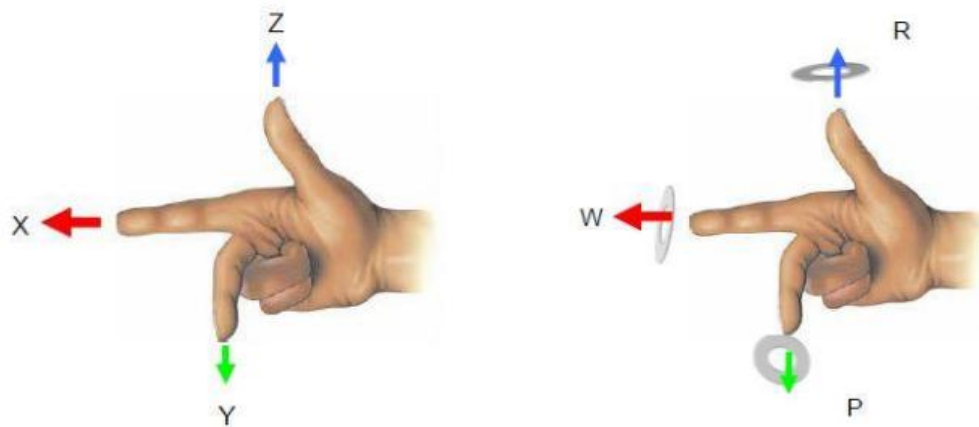
A robotkar programozásához és programíráshoz egy érintőképernyős TEACH Pendant használhatam.



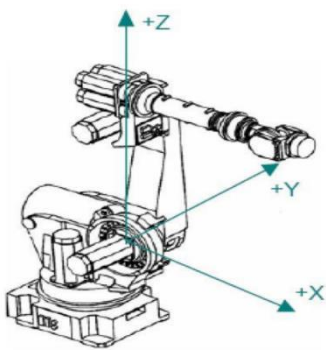
Szeretnék kitérni pár fontosabb alap robotkezelési és programozási lehetőségekre, ami fontos a ha robotokat használunk.

### **1. Koordináta rendszerek:**

A koordináta rendszerek a pozíciók helyzetének és orientációjának leírására használjuk. A robotkarok 4 féle koordinátarendszert ismer (mind a 4-nél különbözően mozog a robot).

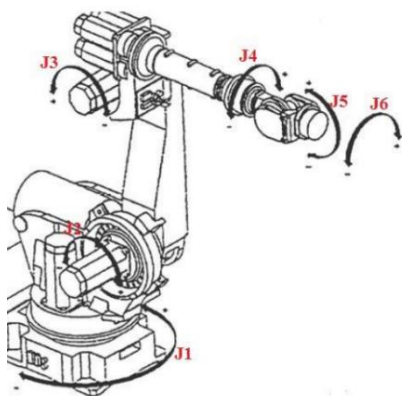


1. ábra Jobbsodrású XYZ koordináta és WRP való elforgathatóságuk (Fanuc robot programozás kézikönyvből szerkesztve)



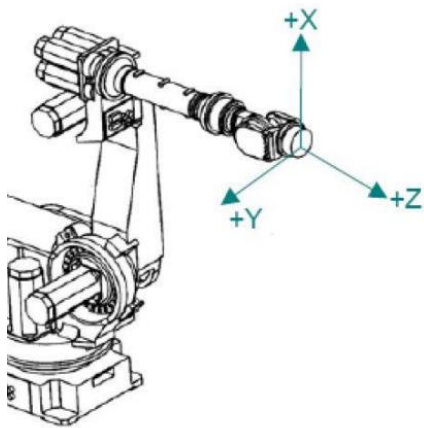
2. ábra World Fream Fanuc\_Robot\_KAH szerkesztve

Világ koordináta rendszer(World Frame) a robot alapértelmezett koordináta rendszere. XYZ irányú mozgások alapértelmezetten a Világ koordinátában történik. Derékszögű jobbsodrású rendszer. (Descartes-féle). A nullpontja általában az első tengely középpontjában található. A robotkar végközéppontja XYZ síkon történő mozgáskor síkon marad, amit tengelyek egymással összehangoltan mozgatásával oldja meg.



3. ábra Joint Coordinate System Fanuc\_Robot\_KAH szerkesztve

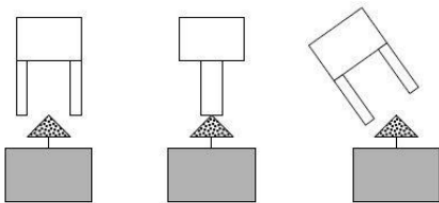
Ízületi koordináta rendszerben (Joint Coordinate System) tengelyenként tudjuk mozgatni a robotot. Minden tengelynek van nullpontja, amit jeleznek is általában robottesten, hogy ízületenként hol található. Ez a robot koordináta rendszerében elhelyezett pozíciók helyzeteinél lesz fontos. A World Frame-vel együtt csak ez a kettő alapértelmezett koordináta rendszere. Ami azt jelenti, hogy nem lehet ezen koordinátarendszerek tulajdonságait megváltoztatni. Word Frame-ben a koordináta rendszer origópontját és az XYZ síkok irányait, míg az Ízületi koordinátában pedig a tengelyek elhelyezkedéseit.



4. ábra TOOL FRAME Fanuc\_Robot\_KAH szerkesztve

Szerszám koordináta rendszer (Tool Frame) World Frame-hez hasonlóan ez is jobb sodrású, de a koordináta rendszer nullpontja nem az első tengely középpontjában található, hanem a csatlakozó felület középpontjában, de ez áthelyezhető a ráhelyezett szerszám csúcsára is, amit TCP (Tool-Centre-Point) nevezünk. Ezen koordináta rendszerben való mozgás során a szerszám nullpontja (TCP) és a robotcsuklóra meghatározott szerszám-koordináta rendszer alapján fog mozogni. A TEAch Pandentben 10 darab Tool Frame-t tudunk regisztrálni és azok között váltani. TCP pont beállítására két plusz egy féle

lehetőség van.

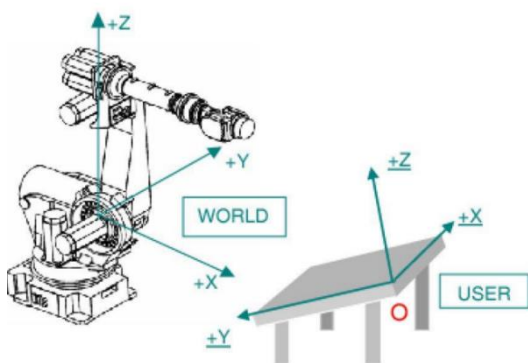


5. ábra TCP betanítása 3 pontos eljárással Fanuc\_Robot\_KAH szerkesztve

Van a három pontos eljárás, amely során a pontot három különböző irányból megközelítjük és ezt tanítjuk meg a robotnak.

Hat pontos eljárás, amely során hasonlít a három ponthoz csak, itt további három pont megadása szükséges.

Közvetlen megadás: A koordinátarendszer közvetlen megadását, számszerű felvételét jelenti.



6. ábra USER FRAME koordinátarendszer Fanuc\_Robot\_KAH szerkesztve

A felhasználó koordináta rendszer (USERFRAME) Descartes.féle koordináta rendszer. A felhasználó által megadott koordináta rendszer az alap koordináta rendszerből van kalkulálva és bárhol lehet a térben. A pozíció adatok az eszköz pozícióját határozzák meg a megadott koordináta rendszerhez viszonyítva. A USERFRAME bemérése szükségünk van a koordináta tengelyek origó pontjára, X és Y tengelyek irányaira.

## 2. Program létrehozása:

Ahhoz, hogy egyáltalán tudjunk programot írni vagy mozgatni a robotot, szükségünk van egy üres munkalapra. A TEACH Pendant monitoron amikor bekapcsoltuk a robotot és felállt a rendszer, akkor megjelenik alapmenü, ahol képesek vagyunk új programot készíteni. Amibe megadhatjuk program nevét, maga a program típusát, hogy programot vagy macrot készítsünk. Továbbá az új munkánkat tudjuk kommentezni is, ha számos programunk van elmentve a tárhelyben. Egy programba képesek vagyunk pozíciókat megadni, amit a robot feltud venni és logikai műveleteket, amit a robot képes elvégezni.

## 3. Robotkar mozgásutasítása

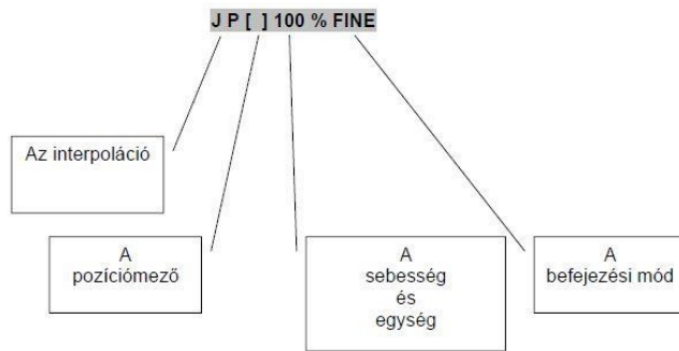
Miután elkészítettük egy üres munkalapot, képesek vagyunk robotkart kézi vezérléssel mozgatni és felvenni azon pontokat a robotkar munkaterében, amit szeretnénk felhasználni a program írás során. Azt, hogy a robotot, milyen elv szerint mozgatjuk és milyen gyorsan mozogjon, a kiválasztott koordináta-rendszer és sebesség határozza meg.



7. ábra Teach Pendant való kézi mozgatáshoz szükséges gombok (FANUV\_Robot\_KAR szerkesztve)

Amikor robotkarral bizonyos pozíciókat jegyeztettünk meg akkor a munkalapon program-sorok jelennek meg. A programsor olyan különböző elemekből áll, amelyek a robot két pozíció közötti mozgásának módját határozza meg. Az elemek megváltoztatásával a robotot a kívánt módon, a megfelelő pályán mozgathatjuk. Egy programsor egy mozgást vezérlő utasítás (J;L;C), pozíció szám (P[]), sebesség (mm/sec;%;stb.), és mozgásfajta (FINE;CNT0-100) programelemekből áll. További program elemekkel is tudjuk egy programsort bővíteni, de ezeket alapból nem jelennek meg nekünk kell külön elhelyezni. Ilyen például a gyorsulás (ACC0-150)

Egy programsor alap kinézete:



8. ábra Alap programsor összetétele. Saját szerkesztés

Ez álltal képesek vagyunk robot számára kigondolt munkának vázát létrehozni, amit különböző logikai műveletekkel képesek vagyunk tovább optimalizálni.

Példa: Egy tárgynak A pontból B pontba vitelének mozgás utasítása programja

J P [1] 100% CNT10(tárgy fölé állás)

L P [2] 2000mm/sec FINE(tárgyra való mozgás, A pont)

L P [1] 100% FINE(tárgy kiemelése)

J P [3] 100% CNT40 ACC150%(B pont fölé állás)

L P [4] 100% FINE(B pontra való lehelyezés)

A mozgás vezérlő meghatározza, hogy a robot milyen pályán mozog két pont között. Amit a program sor legelején találjuk meg. Három féle pálya típust különböztettünk meg. A Joint-os, Linear-os és a Circular-t.



Joint-os mozgás típusnál a robotkor leggyorsabb úton halad a két pont között. Ezt a féle utasítást akkor használjuk, ha nem számít, hogy a robot kar milyen pályát ír le. Ennél a mozgásnál a tengelyek egyszerre mozgatja és állítja be a kívánt pozícióba.



9. ábra Joint-os mozgás. Fanuc\_Robot\_KAH szerkesztve

A Linear mozgásvezérlő utasításánál a TCP pont a két pont között egy egyenes mentén fog mozogni. A TCP pont a pálya mentén fokozatosan áll át a kezdőpont orientációjába.



10. ábra Linear mozgás. Fanuc\_Robot\_KAH szerkesztve

Az utolsó pedig a Circula mozgásvezérlő utasítás a TCP pont egy körív mentén halad a kezdőpontból egy közbenső ponton keresztül halad a végpontba.



11. ábra Circular mozgás. Fanuc\_Robot\_KAH szerkesztve

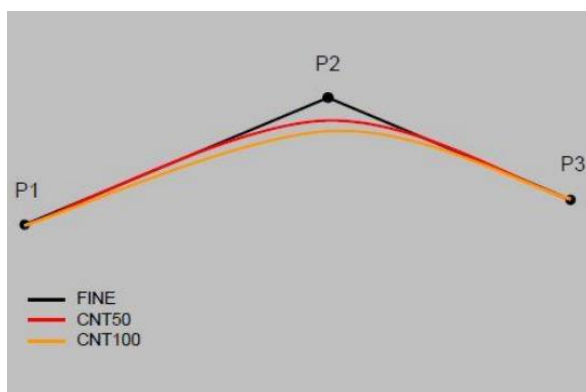
A pozíciómező (P[]) program elem alatt olyan információk rejlenek hogy az adott pozíció koordinátái XYZ és WPR ezenfelül a pozíció felvételekor használt User frame és User Tool is megtudhatjuk.

Sebesség programsor elemmel tudjuk a két pont közötti sebességet változtatni. Maximális sebességnél nem tudunk nagyobb értéket megadni. Különböző mozgásvezérlő elemekhez különböző lehetőségeink vannak a mértékegység megadás formátumával kapcsolatban. Jo-int a mozgás típusnál, százalékos vagy sec (másodperc) mértékegységek választhatók. A százalékos megadásával a tengelyek maximális sebességéhez viszonyítja. Míg a sec megadásával azt állítjuk be, hogy a két pozíció közti utat milyen gyorsan tegye meg.

A másik kettőnél Linear, vagy Circula-nál ugyanazok sebesség mértékegységek használhatóak, amik lehetnek mm/sec, cm/min, inch/min, deg/sec vagy sec mértékegységek adhatók meg.

Az utolsó programsor elem Mozgásfajta elem. Két féle mozgás mód van, FINE vagy CNT. Fine mozgásfajtnál robot pontosan érint minden beállított pontot, ahol egy pillanatra megáll, majd halad a következő pontra.

CNT mozgásfajtnál Robot útba ejt minden pontot, de nem áll meg és nem féltetlenül érinti a pontot pontosan. A folyamatosság értéke meghatározza mennyire közelíti meg a pontot. Ez az érték 0-100 között veheti fel, amiben a 100 érték a robot a legmesszebb minimális lassítással mozog.



12. ábra Fine és a CNT mozgás összehasonlítás. Fanuc\_Robot\_KAH szerkesztve

Program sor végén képesek vagyunk további de alpból nem megjelenő programsor elemeket beszúrni. Ilyen programsorelem például gyorsulás elem (ACC). Ezen elem segítségével növelni vagy csökkenteni lehet a robot gyorsulási és lassítási idejét, amit százalékban adhatunk meg 0% és 150% között. Példa 50% értéknél az átlagos gyorsulás/fékezés kétszer annyi időbe telik mire eléri a végleges sebességét.

Ezen egyszerű mozgató Programoknak a programsorait képesek vagyunk szerkesztésére, formázására.

1. Insert(beszúrás): üres programsort hozz létre
2. Delete(törlés): törli a programsort
3. Copy and Paste(másolás): programsor másolja és beillesztése
4. Find (Keresés): Rákeres bizonyos program sorokra
5. Replace(Kicserélés): A program sorokat
6. Renumber (Újra számolás): A pozíció mezőket újra számolja fentről lefele
7. Comment (Komment): Programsorokba képesek megjegyzéseket írni.
8. Undo (Visszavonás): Visszavonja programszerkesztéses feladatokat

Ezen funkciók segítségével képesek vagyunk robot mozgására, de ezen felül a robotkart vezérlő vezérlőszerv nem csak a robot mozgására, hanem logikai- és utasítás műveletekre is képes. A logikai műveletek alapja a FANUC robotkarnál az úgy nevezet regiszterek.

#### 4. Regiszterek:(R[ ]=):

A regiszterek legegyszerűbb megértéseként úgy kell felfognunk, mint ha tárolóként funkcionálnak. Minden olyan információt, amit szeretnénk felfogni a program lefutásakor, egy regiszteren keresztül a megfelelő logikai művelttel képesek vagyunk, de egy általunk megadott értékeket is képes tárolni, addig amíg azt meg nem változtatjuk.

Két remek példa erre a regiszterekett számlálóként való használata és a regiszter nullázása  
 $R[1] = R[1] + 1$  :Minden egyes ciklusonként amikor ide ér a robot +1 hozzá ad a Regiszter[1]-hez.

$R[1] = 0$  : Az R[1] regiszternek az értékét 0-ra állítjuk manuálisan.

### Argument Regiszterek:

Argumentek és Argument regiszterek használatával két program között adatokat tudunk mozgatni. Csak két program között mozoghatnak. Jelölésre az AR[]-t használjuk. Általában akkor használjuk, ha egy fő programból egy mellék programba akarunk valami információt küldeni

### Pozíció Regiszter:

A X,Y,Z,W,P,R formában tárolják egy pozíció információit. A Pozíció regiszterek meg tudják változtatni a program lefutásának az útvonalát. A programon kívül vannak tárolva, így több programban is használhatjuk.

Gyakori felhasználása, ha van egy olyan pozíció, amit gyakran használunk akkor azt eltároljuk egy Pozíció regiszterbe. Így aztán különböző programokban meghívható válik. További előnye, hogy csak a regiszterben kell megváltoztatni a pozíciót és a többi másik programban meghívott regiszter is módosul.

### LPOS/JPOS:

Ezekkel a parancsokkal eltárolhatja a robot pillanatnyi helyzetét egy pozíció regiszterbe. A LPOS lineáris koordinátait tárolja(X,Y,Z,W,R,P), míg a JPOS a robot ízületi (J1...J6) koordinátáit. tárolja el.

## 5. Kimenetek és bemenetek. (Output, Input)

Az I/O nem más, mint egyszerű eszköz a jelekkel történő kommunikációra más eszközökkel. Ha a robot nem önállóan működik, hanem perifériás eszközökkel, például detektorokkal, kamerákkal, más robotokkal, akkor szükségünk van az I/O-ra. A robot információkat kaphat az érzékelőktől, kameráktól vagy más eszközöktől az Input segítségével, és az Output segítségével utasítást adhat motoroknak, más robotoknak

A bemeneteknek és kimeneteknek (I/O) 4 fajtája van:

1. Digitális(I/O): biztosítják az adatok áramlását a ki- és bemeneti vonalokon, jel lehet ON vagy OFF
2. Robot(RI/RO):ki- és bemeneti jeleit a robot és kontroller közötti kommunikációra használják, jel lehet ON vagy OFF.
3. Analóg: a ki- és bemeneteknek folyamatos ki- és bemeneti jelek vannak. Pl.: hőmérséklet, feszültség.
4. Csoportos(GI/GO): egy csoportba tartozó különböző ki- és bemeneti jelek. Bináris számként értelmezzük őket.

## 6. Utasítások:

If Statement (feltételes végrehajtás)

Bizonyos feltételeknek együttesen teljesülni kell, hogy a program utasítást végrehajtsa. Ha a beállított feltételek teljesülnek, a robot meghívhat egy alprogramot, a program megjelölt részére ugorhat, kimenetek vagy bemenetek állapotát változtathatja ON vagy OFF-ra, de akár regiszterek értékét is tudja változtatni.

```
IF R[1] = 3 CALL (PROGRAM NEVE)
```

Ennek egy további lehetősége, ha mondjuk egy regiszterhez több feltételt szeretnénk hozzácsatolni ezt vagy további IF utasításokat hozzunk létre, de van lehetőségünk a Select utasítást választani amivel azt az utasítást fogja választani ami igazgá válik. Például csoportos ki- és bemeneti jelek vizsgálata.

1. IF GI[1]=2 CALL PROGRAM1	1. SELECT GI[1] = 2 CALL PROGRAM1
2. IF GI[1] = 4 CALL PROGRAM1	4 CALL PROGRAM1
3. IF GI[1] = 8 CALL PROGRAM1	8 CALL PROGRAM1
4. IF GI[1] = 16 CALL PROGRAM1	16 CALL PROGRAM1

WAIT utasítás

A WAIT utasítás során a program futásának szüneteltetése vagy feltételre való várásáig a robotot megállítjuk. A robot időtartamra ki vagy bemeneti jelre a regiszter vagy egy megadott értékére várhat.

WAIT 6 (sec) Sima várakoztatás

WAIT DI[1] indító gomb = ON Addig nem halad tovább a programban míg a DI[1] ON-ra át nem vált

LABELS & JUMP LABELS utasítás.

A LABEL & JUMP LABELS utasítással képesek vagyunk egy programon belül ciklusok létrehozására. A LABEL-ek (LBL, címkék), egy pontot jelölnek a programban, ahova a robot ugrik, amikor a JUMP LABELS (JMP LBL, ugrás a címkére) utasítást használjuk. Ezért használhatók ciklus létrehozására.

LBL[1]

J P [1] 100% CNT10(tárgy fölé állás)

L P [2] 2000mm/sec FINE(tárgyra való mozgás, A pont)

L P [1] 100% FINE(tárgy kiemelése)

J P [3] 100% CNT40 ACC150%(B pont fölé állás)

L P [4] 100% FINE(B pontra való lehelyezés)

JMP LBL [1]

A fenti példában egy feltétel nélküli ciklust látunk. Mindentől függetlenül újra meg újra le fog futni. Feltételes ciklus létrehozásához az IF utasítást és egy regiszter számlálót kell kombinálni.

A programban bárhova írhatunk LMP LBL utasítást, hogy bármelyik másik címkére (LBL) átugorjon a programban, még ha az a JMP LBL utasítás után is következik a programban.

#### CALL utasítás

A CALL utasítással egy programon belül egy másik programot tudunk meghívni. Az utasítást tudjuk magába de IF utasítással valamilyen feltételhez is tudjuk kötni.

CALL(program neve)

#### TIMER utasítás (időzítők)

A TIMER utasítással a program egészét vagy egy részét tudjuk időzíteni. A TIMER utasításnál arra figyelni kell, arra Az időzítő eltárolja az utoljára rögzített időt, még a kontrollér kikapcsolása után is. A következőhasználatkor a számlálást innen folytatja Így esetleg nem ad pontos információt egy mérni kívánt folyamatról. Ezért érdemes az időzítőket használat előtt le nullázni, resetelni. Ez az utasítás három részből áll egy RESET START és STOP-ból.

TIMER[2] = RESET

TIMER[2] = START

J P[1] 100% FINE

J P[2] 100% FINE

L P[3] 2000mm/sec FINE

TIMER[2] = STOP

#### VERRIDE utasítás

Programutasításon keresztül tudjuk a robot mozgás sebességét szabályozni. Az érték amit felvehet az 0-100% között lehet

J P[1] 100% FINE

J P[2] 100% FINE

L P[3] 2000mm/sec FINE

VERRIDE 50%

J P[4] 100% FINE

J P[5] 100% FINE

L P[6] 2000mm/sec FINE

FOR utasítás

Egy mozgási sorozat, vagy egyéb utasítások többször ismételt végrehajtására alkalmas utasítás. Használatához szükségünk van egy regiszterhez amiben számoljuk megtett ciklusokat. Azt, hogy hányszor ismétélje meg konstansként regiszterként vagy argument regiszterként tudjuk megadni.

FOR R[1] = 1 TO 10

J P[1] 100% FINE

J P[2] 100% FINE

L P[3] 2000mm/sec FINE

J P[4] 100% FINE

J P[5] 100% FINE

ENDFOR

Makrók

Egy makró utasításnak az a funkciója, hogy egy utasítások sorozatából álló programot egyetlen utasítássá fogjon össze, és ezt a parancsmondatot a végrehajtás kérése után előhívja. Egy makró nem tartalmazhat mozgás utasítást. A makró utasítást egy programban program utasításként tudjuk használni. de képesek vagyunk a TEACH PENDANT egyik gombjára kihelyezni és annak megnyomásával elindítani. Makrókat leginkább a robotkarra felhelyezett beavatkozók működtetésére használjuk például a megfogó zárása és nyitására.

1: RI[1: Centralzár]=OFF

2: WAIT 0,5 (sec)

3: RI[1: Centralnyit]=ON



## **5. Palettázó robot programozása**

Két féle megoldást tudtam kitalálni a palettázó program megvalósítására. Az alap mozgás utasítás, amit feladat megoldására írtam, mint egyfajta segítség, amit tudtam segítségül venni.

1:J P[1] 100% CNT20      Tárgy fölé állás

2:L P[2] 200mm/sec FINE Tárgyra állás

3: Czar      megfogni a tárgyat

4:L P[1] 200mm/sec FINE felemelni a tárgyat

5:J P[3] 100% CNT20      Átvinni a másik tároló fölé

6:L P[4] 200mm/sec FINE behelyezni a tárolóba

7: Cnyit      elengedni a tárgyat

8:J P[3] 100% CNT20      vissza emelni a robotkart

Az egyik megoldás, hogy végig felveszem a pontokat és a programom csak pozíciókból és Czar és Cnyit macrokból állna. Ami rengeteg munka könnyen ellehet veszni a program vissza olvasásában, hogy melyik pozíció hova tartozik, amit megelőzve rengeteg kommentet kellene írni a követhetőség szempontjából és egyáltalán nem rugalmas. Így a program csak erre feladatra lenne képes, ha már egy kicsit is elmozgatjuk a henger tárolókat akkor teljesen újra kéne venni a pontokat.

A másik lehetőség, hogy paraméteresen oldom meg ami alapján csak egy pozíciót fogok felvenni és mindent ahhoz fogok viszonyítani. Annak az egy koordinátájának változtatásával fogom meghatározni az összes pozíciót, ami jóval nehezebb feladat programozás szempontjából, mint az előző, de sokkal rugalmasabb, mint a másik. Ha valami változás történik nem kell teljesen újra írni a programot pár változtatás végezve, mondjuk a UserFrame újra felvétele vagy sorok oszlopok számának megváltoztatása, és újra tudjuk használni a programot. Ezért inkább ezt választottam. mint megoldás menet. amit most a szakdolgozatom fent maradó részében szeretnénk részről részre bemutatni.

A palettázó program elején a kezdő értékeket, inicializálást hajtottam végre a 9.sorig, mint hogy a robot melyik USERFRAME, USERTOOL- t használja, a kimenetek kikapcsolása és az alapsebesség beállításával foglalkoztam.

- 1: UFRAME\_NUM=4 ;
- 2: UTOOL\_NUM=2 ;
- 3: DO[501:kek led]=OFF ;
- 4: DO[502:feher led]=OFF ;
- 5: DO[503:feher led2]=OFF ;
- 6: DO[504:kek led2/TECHMAN MEHET]=OFF ;
- 7: OVERRIDE=5% ;
- 8: Cnyit ;
- 9: ;

A 10. sortól pedig már program főrésze található.

A 10 és 11 sorban egy indító programnak megfelelő utasítást olvas a robot, amiben ha a robot automatikusan megy is szükséges legyen egy start gomb mielőtt bármiféle mozgást is végezne. Ezt leginkább biztonsági okokból volt szükséges betenni a programba, balesetek megelőzése szempontjából. Összetétele nagyon egyszerű WAIT utasítás (várj) amit a digitális jel ON-ra kapcsolva tovább lépjen a 11 sorba, ahol ugyan ennek digitális jelnek a kimenete is kapcsoljon be ezzel tudtára adva a felhasználónak, hogy a robot most már mozogni fog.

- 10: WAIT DI[501:kek gomb]=ON ;
- 11: DO[501:kek led]=ON ;

A 12 és 15 sorban egy JUMPLABEL utasítás látunk, ami két felvett pozíciót helyet ugor át. Ez két pozíció, amit körbe fog, a raklap egymással szemben lévő két saroknak a pozícióját rejti. A két pozíciónak majd később lesz fontos szerepe.

12: JMP LBL[1] ;

13:L P[1] 200mm/sec FINE ;

14:L P[2] 200mm/sec FINE ;

15: LBL[1] ;

A 16 sorban a robotot alaphelyzetbe állítjuk.

16:J PR[1:HOME] 50% FINE ;

A programmal beolvassuk, hogy a PR[9]-ben raktározott információkat.

17: OFFSET CONDITION PR[9:Egyseg\_elt] ;

A 18 és 19 sorban regiszterben megadott sor és oszlop számokat adjuk meg hogyan szeretnénk helyezni a formákat. Ehhez pedig egy másik regiszterben ehhez az értékhez kivontunk egyet, ami a formák közötti távolságok kiszámolásához és a FOR ciklusoknál lesz szükségünk.

18: R[7:Palle\_s]=R[5:Palle\_s-1]-1 ;

19: R[8:Palle\_o]=R[6:Palle\_o-1]-1 ;

A 20 és 21 sorban a két felvett pozíciónak koordináta értékeit két pozíció regiszterbe helyezzük.

20: PR[4]=P[1] ;

21: PR[5]=P[2] ;

Majd egy másik pozíció regiszterben kivonjuk egymásból így megkapjuk azon számokat, amivel képesek vagyunk a programunkat paraméteresen megoldani.

22: PR[6:PARAMETER]=PR[5]-PR[4] ;

A 23 sorban egy másik pozíció regiszterbe töltjük fel P[1] koordinátáit ami az első pont ahol szeretnénk az első tárgyat megfogni. A PR[7] regiszterben fogjuk paraméteresen változtatni a koordinátáit amivel létrehozuk a paletázás műveltet.

23:  $PR[7:ELMEG\_pont]=P[1]$  ;

24 és a 25 sorban létre hoztam egy másik paraméteresen változtatható pozíciót aminek a feladata hogy a robot kar mindig fölé álljon a megkívánt fogni tárgy felé és ahova majd szeretnénk letenni. A 25 sorban a  $PR[8]$  harmadik koordinátájához más szóval a Z-koordinátájához hozzá adtam 60.

24:  $PR[8:FEL\_pont]=P[1]$  ;

25:  $PR[8,3:FEL\_pont]=PR[8,3:FEL\_pont]+60$  ;

A 26 és 27 sorban pedig X és Y tengelyen való léptetés értékét számolom ki, amit egy regiszterben tárolom.

26:  $R[20:X-tav]=PR[6,1:PARAMETER]/R[6:Palle\_o-1]$  ;

27:  $R[21:Y-tav]=PR[6,2:PARAMETER]/R[5:Palle\_s-1]$  ;

28: ;

A FOR ugyan úgy egy ciklust tudunk létrehozni annyiszor fogja ciklust újra indítani amíg megcsinálja 0-tól a  $R[6]$  megadott értéket el nem éri. minden egyes ciklus lefutása után hozzá ad egyet a  $R[2\_értékhez]$ .

29:  $FOR R[2:x\_ertek]=0 TO R[6:Palle\_o-1]$  ;

Azzal hogy az aktuális koordináta értékével kivonást végzünk így a  $PR[9]$  regiszternek a koordinátáit nullázni tudjuk. Így ebben a regiszterben kitudjuk számolni újra és újra hogy az alap ponttól mennyivel kell hozzáadni ahonnét felvesszük a következő tárgyat.

30:  $PR[9:Egyseg\_elt]=LPOS-LPOS$  ;

A 31. sorban a  $PR[9,1]$  első koordináta rendszerét számoljuk az X irányú elmozdulást. A  $R[20]$  tárolt egységnyi távolságot a tárgyak között és a FOR ciklus által számolt  $R[2\_érték]$  szorzatával.  $PR[9] = 24,77 * 1 = 24,77$  akkor ennyivel fogja X irányban eltolni az alap pozíciótól.

31:  $PR[9,1:Egyseg\_elt]=R[20:X-tav]*R[2:x\_ertek]$  ;

A 32. és 33 sorba ugyan azt viszük végig mint a 29. és a 31. sorban csak itt az Y irányban számolom az eltolni kívánt mennyiséget.

```
32: FOR R[3:y_ertek]=0 TO R[5:Palle_s-1] ;
```

```
33: PR[9,2:Egyseg_elt]=R[21:Y-tav]*R[3:y_ertek] ;
```

A 34. sortól egészen 39. sorig az USERFARME váltását hozom létre. Ezzel oldom meg hogy a felvett tárgyat a másik pozícióba vigye. Ugyan úgy kezdem egy FOR ciklussal aminek értékét a R[9] regiszterben tárolom. Ez által hogy a regiszterben 0 és 1 érték lehet egy feltétel tudok készíteni amiben az szerepel ha a R[9] értéke 0 akkor USERFRAME\_NUM=4-ba mentet koordináta rendszert ha nem ez az értéke akkor az USERFRAME\_NUM=5 mentet koordináta rendszert használja.

```
34: FOR R[9:USER_switch]=0 TO 1 ;
```

```
35: IF (R[9:USER_switch]=0) THEN ;
```

```
36: UFRAME_NUM=4 ;
```

```
37: ELSE ;
```

```
38: UFRAME_NUM=5 ;
```

```
39: ENDIF ;
```

```
40: ;
```

```
41: ;
```

A 42. sortól 47. sorban a Z vektor körüli elforgatása történik meg. Erre azért van szükség, mert amikor a megfogott hengert le szeretném tenni a másik pozícióba, akkor bele ütközött a robot megfogója, a mellé letett hengerbe. Erre azt a megoldást találtam ki, hogy a megfogó pofák iránya ne legyen párhuzamos lerakási sorrend irányával. Így 30° elforgatással nem ütközött bele más tárgyba. Ugyan úgy egy pozíció regisztert használtam, aminek az értékét ki nullázom a 42. sorban. Attól függően, hogy a robot melyik felhasználói koordináta rendszert használja hozzá ad vagy kivon a Z tengely körüli forgatás értékéből 30-at.

```
42: PR[10:Egyseg_elf]=LPOS-LPOS ;
```

```
43: IF (R[9:USER_switch]=0) THEN ;  
44: PR[10,6:Egyseg_elf]=PR[10,6:Egyseg_elf]+30 ;  
45: ELSE ;  
46: PR[10,6:Egyseg_elf]=PR[10,6:Egyseg_elf]-30 ;  
47: ENDIF ;
```

A 48. sortól a 55. sorig két led lámpa felkapcsolása és lekapcsolásán logikája van az alapján, hogy a robot éppen melyik ponthoz megy vagy tartózkodik. Ha annál pontnál van ahol a tárgyat felveszi vagy éppenséggel oda tart akkor a fehér led van felkapcsolva és a kék led van lekapcsolva. Ha pedig éppen a másik pozícióba halad vagy készül letenni a megfogott tárgyat, akkor ledek pont fordítva a fehér le-, míg a kék led felkapcsolva van. Erre egyszerű IF utasítást alkalmaztam, amiben a USER\_switch állapotát vizsgálom hogy 0 vagy 1 és ez alapján a program kapcsolgatja a ledlámpákat fel és le.

```
48: IF (R[9:USER_switch]=0) THEN ;  
49: DO[503:feher led2]=ON ;  
50: DO[504:kek led2/TECHMAN MEHET]=OFF ;  
51: ENDIF ;  
52: IF (R[9:USER_switch]=1) THEN ;  
53: DO[503:feher led2]=OFF ;  
54: DO[504:kek led2/TECHMAN MEHET]=ON ;  
55: ENDIF ;
```

A 56. és a 58. sorban a két pozíció regiszternek a PR[9] és PR[10] számolt eltolás és megadott elfogadás értékeket hozzáadjuk. Majd mikor a robot kar felvennie vagy letenné tárgyat egy programba ágyazott program utasítással alacsonyra vesszük a sebességét

```
56:J PR[8:FEL_pont] 25% FINE Offset,PR[9:Egyseg_elt] Tool_Offset,PR[10:Egyseg_elf]
```

```
57: OVERRIDE=5% ;
```

```
58:L PR[7:ELMEG_pont] 200mm/sec FINE Offset,PR[9:Egyseg_elt]  
Tool_Offset,PR[10:Egyseg_elf] ;
```

A 59.-től a 68.-ig egy további feltétel utasítást látható ami a megfogó pofák állásáért felelnek az alapján hogy a R[9] értéke 0 vagy nem. Ha 0 akkor zárja a megfogó pofákat, ha más akkor nyissa ki. A Czar és Cnyit egy macrók amik robot digitális jeleket tartalmaznak. A 64. sorban lévő feltétel utasítás pedig ugyan úgy egy ledlámpa fel- lekapcsolásáért felel, amiben jelzi, hogy a robotkar kezében van e tárgy van nincs.

```
59: IF (R[9:USER_switch]=0) THEN ;
```

```
60: Czar ;
```

```
61: ELSE ;
```

```
62: Cnyit ;
```

```
63: ENDIF ;
```

```
64: IF (R[9:USER_switch]=0) THEN ;
```

```
65: DO[502:feher led]=ON ;
```

```
66: ELSE ;
```

```
67: DO[502:feher led]=OFF ;
```

```
68: ENDIF ;
```

A 69. sorban visszaáll lineáris pályán ahonnét megközelíti a felvenni kívánt tárgyat vagy ahonnét letenné kezdő pozíciójába, amit ugyan úgy egy OFFSET és TOOL\*OFFSET utasítással oldottam meg, mint ahogy 58.sorban van. 70 sorban az OVERRIDE utasítással felgyorsítom a robot mozgását.

```
69:L PR[8:FEL_pont] 200mm/sec FINE Offset,PR[9:Egyseg_elt]  
Tool_Offset,PR[10:Egyseg_elf] ;
```

```
70: OVERRIDE=40% ;
```

```
71: ENDFOR ;
```

```
72: OVERRIDE=40% ;
```

```
73: ENDFOR ;
```

```
74: ENDFOR ;
```

Miután végig ment az összes FOR cikluson vissza tér a PR[1] tárolt HOME pozícióba és az indító gombnak a fényét lekapcsolja.

```
75:J PR[1:HOME] 5% FINE ;
```

```
76: DO[501:kek led]=OFF ;
```

A műveletről video anyagot is készítettem, amit az alábbi linken meglehet tekinteni a robot munkavégzés közben.

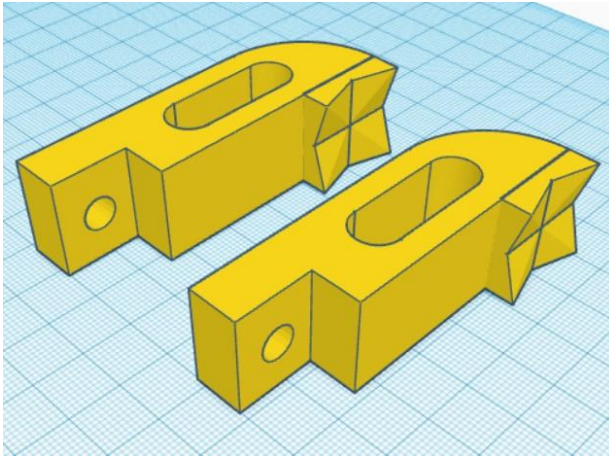
<https://drive.google.com/file/d/15A5TuNEQ7VN4xhtEilcCPy1X2TYLV5Zl/view?usp=sharing>

<https://drive.google.com/file/d/1FbLqvOhWV9ZBjX97VbZesTvtPqeW4wFg/view?usp=sharing>



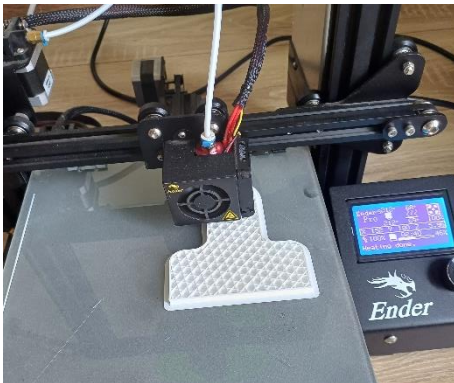
Arról is szeretnék írni, milyen problémákba nehézségekbe ütköztem a programozás közben.

Először is a szakami gyakorlaton, nem volt olyan megfogó pofa, ami alkalmas lett volna a henger test a megfogására. Ezért nekem kellett terveznem két megfogó pofa darabot, amit felszerelve a robotra tudom a berendezést hengerek paletázására használni. Megtervezés szempontjából fontos volt, hogy a megfogó, amivel lehetőségem volt dolgozni (SCHUNK



13. ábra Megfogópofa 1 prototípus Tinkercadben. Sajátkép

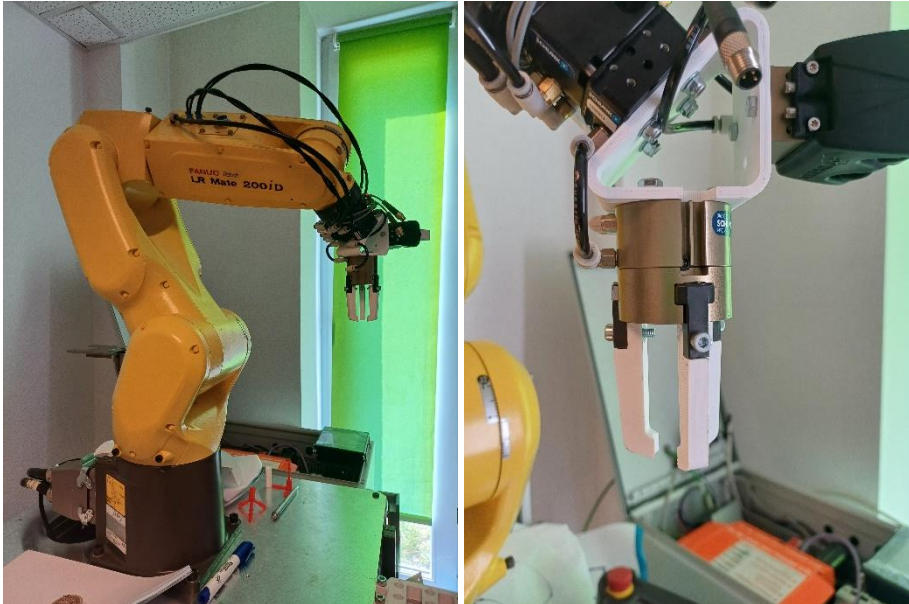
MPZ típusú centrál megfogó) arra kompatibilis legyen továbbá kellően hosszú. Ezenfelül, mikor megfogó szerkezete összeszár a pofák pontosan illeszkedjen a hengertest köre. A pofa megszerkesztésére egy ingyenesen használható internetes alkalmazást használtam a Tinkercad-et amiben egyszerű geometria testekből sikerült egy megfogó pofát létrehozni.



14. ábra Saját tervezésű külső enrgilánc tartó 3D nyomtása közben. Sajátkép

Majd ezt egy ottani 3D nyomtatóval és ugyan úgy egy ingyenes programot a Creality használva ki is nyomtattam. Amiben az ottani munkatársak segítettek, hogy milyen beállítással érdemes az én megfogómat ki nyomtatni. Itt több kísérletezés és modell megváltatása után sikerült két olyan pofát kinyomtatni, ami minden feltételnek megfelel.

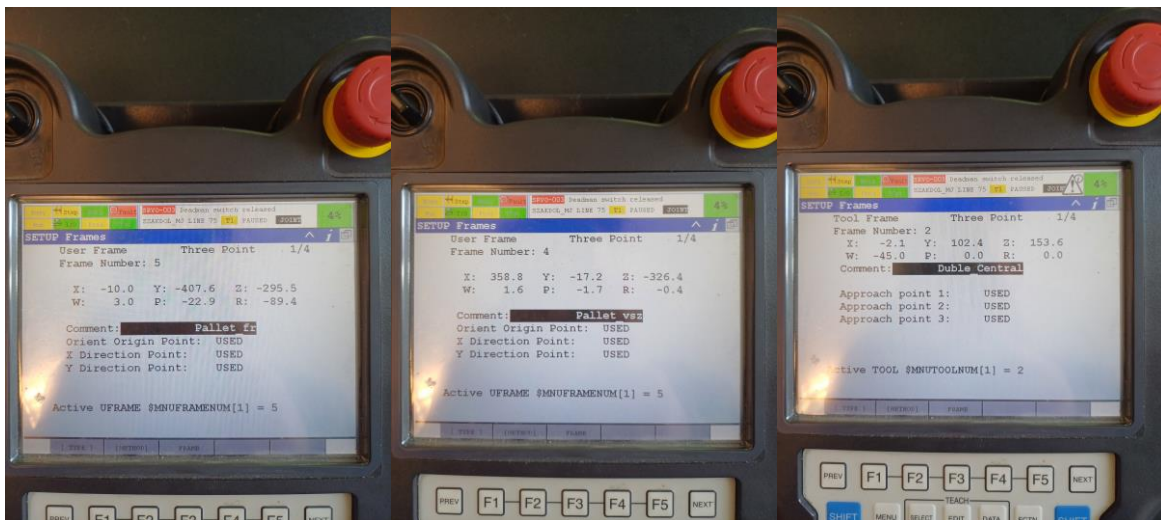
Ezután foglalkoznom kellett azzal a feladattal, amiben megfogott össze kellett kötnöm a robottal a pneumatikus csövekkel és letesztelni, hogy jól kötöttem be. Csak is ezután tudtam elkészíteni a paletázó programot.



*15. ábra pneumatikus csövekkel felszerelt és összeszerelt Fanuc robot és megfogó közeli képe a felszerelt végső pofákkal. Saját kép.*

A program írás közben az két legnehezebb része volt a USERFRAME-ek és a TOOLFRAME beállítása. Miután a feladatomat paraméteresen oldottam meg így nagyon fontos volt, hogy a használt USERFRAME-eket a lehető legpontosabban állítsam be, mert minden pozíciót ez alapján akarom meghatározni, ha pontatlan vagyok akkor a henger testeket nem fogja tudni a robot kar pontosan megfogni és pontosan letenni. Minél nagyobb különbség van az eltolásiérték és a valós érték között annál hamarabb fog a henger test fennakadni a hengertest tartó peremén.

Ezért a program tesztelésekor rengeteg időt szántam a használt USERFRAME optimalizálására. A USERFRAME-eket a 3 pontos eljárást alkalmazva hoztam létre a koordináta-rendszer, aminek az egyik sarokban lévő hengertartó közép pontja a koordináta rendszer origó pontja és a sorok és oszlopok pedig az X és Y tengelyeknek felelnek meg. A USERTOOL beállításánál a közvetlenül vittem be a TCP pont koordinátáit, ami alapján el kellett tolnom a robotkar végpontjából a szerszám végpontjába a koordináta-rendszer nullpontját. Miután megtaláltam ezt a pontot utána a XYZ tengelyeket kellett a helyes irányokban elforgatnom, hogy a Z-tengely megegyezzen a szerszám tengelyével.



16. ábra A USERFRAME-ek és TOOLFRAME beállítás monitor képe. Sajátkép

## 6. Összesítés:

A szakdolgozatom elkészítése során sokat tudtam tanulni a 4.0 ipari forradalomról. Amiben részletesebben megértettem milyen részei vannak, mint a Big Data, IoT kapcsolatok és hogy ezek hogyan fonódnak egymásba. Az ipar4.0 belül a robotikával foglalkoztam mélyre hatóbban megismerhettem történelmüket, hogy mai társadalomban számos helyen használják nem csak a nehéz és autóiparban, ahol természetesen azért a legnagyobb mennyiségben használják. De jelen vannak az orvoslásban szórakoztató iparban is. A robotika hihetetlenül megnövelte a termelékenységet a kézi gyártási folyamatokkal szemben. Sajnos élelmiszeriparban még robotokat nem használják nagy számban a robotokat, ami az ipar speciális helyzete miatt. Természetesen raklapozó és csomagoló feladatokat ebben az iparban is elvégeznek. A szakmai gyakorlatom során volt lehetőségem egy ilyen cella megtekintésében, amiben pont

lisztes zsákoknak a palletázása és csomagolása volt a feladat. Még számos technológia nem elérhető, amivel szélesebb körben kihasználható lehetne a robotok potenciálja az élelmiszeriparban. Ezért ez az iparág, ami az egyik legnagyobb fejlesztési potenciállal rendelkezik. A legnagyobb lehetőségek a fejlesztésekben a CPS-tervezés, a HMI-ben, a robottanulási és -képzési szoftvermegoldásokban, a látásrendszerekben, a robotszerkezeti átkonfigurálhatóságban és a robotok karbantartásában van.

Saját tapasztalatom is szeretném megosztani. A SPAR-os szakmai gyakorlatom során, találok olyan munkakörrel a húsfeldolgozó üzemben, amit emberi munkalánccal oldanak meg, de robot vagy kollaboratív robot segítségével, lehetne tovább automatizálni a részleget. Ezáltal kiváltani az emberi munkaerőt a fizikai munkából.

## **7. IRODALMI HIVATKOZÁS**

Benedek G. (2020). Mi az a Big Data és mire használjuk?. Lexunit. <https://lexunit.hu/blog/mi-az-a-big-data-es-mire-hasznaljuk/>

Debrenti, A. S. (2020). A digitalizáció helyzete a magyar élelmiszer-feldolgozó ágazatban. International Journal of Engineering and Management Sciences, 5(1), 196–209. <https://doi.org/10.21791/IJEMS.2020.1.17>

Erdei E., Oláh J., Popp J., (2019). A jövő kihívásai fenntarthatóság és automatizálás. LOGISZTIKAI TRENDEK ÉS LEGJOBB GYAKORLATOK. V. évfolyam, 1 szám. <https://logisztikaitrendek.hu/wp-content/uploads/2019/08/2-OI%C3%A1h-Popp-Erdei-1.pdf>

Koczka P. (2021). Mit jelent az Ipar 4.0, mi jellemzi?. Sensortech Pro. <https://airmonitor.hu/ipar-4-0/>

Kovács Gy. (2015). Robotika és Gyártásautomatizálás. Pécsi Tudományegyetem Műszaki és Informatikai Kar. [https://mik.pte.hu/Attachment/get/portal\\_article\\_attachment/307/kovacsgyorgy-robotikaesgyartasautomatizalas2.pdf](https://mik.pte.hu/Attachment/get/portal_article_attachment/307/kovacsgyorgy-robotikaesgyartasautomatizalas2.pdf)

Ipari automatizálás Robot-x Hungary Kft, RJ3 programozási kézikönyv

Fanuc Educational Robot -HU <https://www.fanuc.eu/~media/files/pdf/products/robots/educational%20cell/fanuc%20educational%20robot%20-hu.pdf?la=hu>

FANUC\_Robot\_KAH <https://drive.google.com/file/d/1RMVcP3BWkY23v5NK8ZVs5B41onQenFXp/view?usp=sharing>

Jamshed Iqbal, Zeashan Hameed Khan, Azfar Khalid (2017) Prospects of robotics in food industry <https://www.scielo.br/j/cta/a/g8Vvv8rW7D7rsfwctx9gmvc/?lang=en#>



## NYILATKOZAT

a záródolgozat/szakdolgozat/diplomadolgozat/portfólió<sup>1</sup> nyilvános hozzáféréséről és eredetiségéről

A hallgató neve: MICHELISZ JÁNOS GYÖRGY  
A Hallgató Neptun kódja: NKZPUG  
A dolgozat címe: ROBOTOK ALKALMAZÁSA AZ ÉLELMISZERIPARBAN  
A megjelenés éve: 2023  
A konzulens tanszék neve: FIZIKA-AUTOMATIKATANSZÉK

Kijelentem, hogy az általam benyújtott záródolgozat/szakdolgozat/diplomadolgozat/portfólió<sup>2</sup> egyéni, eredeti jellegű, saját szellemi alkotásom. Azon részeket, melyeket más szerzők munkájából vettem át, egyértelműen megjelöltem, s 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 tulajdonkezelési szabályzatában megfogalmazottak érvényesek.

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

Kelt: 2023 év november hó 8 nap

  
Hallgató aláírása

<sup>1</sup> A megfelelő dolgozattípus meghagyása mellett a többi típus törlendő.

<sup>2</sup> A megfelelő dolgozattípus meghagyása mellett a többi típus törlendő.

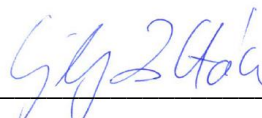
## NYILATKOZAT

Michelisz János (hallgató Neptun azonosítója: NK2PU5) konzulenseként nyilatkozom arról, hogy a diplomadolgozatot áttekintettem, a hallgatót az irodalmi források korrekt kezelésének követelményeiről, jogi és etikai szabályairól tájékoztattam.

A diplomadolgozatot a záróvizsgán történő védelemre **javaslom** / **nem javaslom**.

A dolgozat állam- vagy szolgálati titkot tartalmaz: igen **nem**

Kelt: 2023. év november hó 8. nap



---

belső konzulens