



TUDOMÁNYOS DIÁKKÖRI DOLGOZAT

**BEÁGYAZOTT RENDSZEREK
ALKALMAZHATÓSÁGA GYÁRTÓSORI
BERENDEZÉSEKBEN**

Szerzők: **Losonczy László**
villamosmérnök BSc. szak, III. évf.
Schmidt Péter
villamosmérnök BSc. szak, III. évf.

Konzulens: **Sándor Tamás**
adjunktus

Tartalom

1.	Probléma felvetése.....	3
2.	Célkitűzés.....	4
3.	Rendelkezésre álló eszközök	5
3.1	MPS Állomások.....	5
3.1.1	Munkadarabok	5
3.1.2	Kupakok	6
3.1.3	Handling állomás	7
3.1.4	Cap állomás	11
3.1.5	Robotállomás.....	14
3.2	Mikrokontroller board.....	16
4.	Megvalósított folyamat	19
4.1	A rendszer kialakításának ismertetése	19
4.2	A működtetés rövid leírása	20
5.	Tervezés és fejlesztés	23
5.1	TTL – 24V konverter	23
5.2	SPI port bővítő.....	25
5.3	TTL-RS232 illesztő.....	26
5.4	Kommunikációk	26
5.5	Programfejlesztés	28
5.5.1	Handling állomást vezérlő mikrokontroller programja	28
5.5.2	Cap állomást vezérlő mikrokontroller programja:	30
5.5.3	Robot állomást vezérlő mikrokontroller programja.....	31
5.5.4	PC oldali mikrokontroller programja	32
5.5.5	PC felügyeleti program	34
6.	Megoldás elemzése	36
7.	Továbbfejlesztési lehetőségek	38
8.	Köszönetnyilvánítás.....	39
9.	Irodalomjegyzék	40

1. Probléma felvetése

A TDK dolgozat főtémája a FESTO MPS, vagyis a Modular Production System (Moduláris Gyártási Rendszer) beüzemelése és üzembe helyezése ATMEL mikrokontrollert tartalmazó beágyazott vezérlővel, valamint annak felügyelete egy PC-n futó alkalmazás segítségével.

FESTO MPS állomásokkal nemzetközi szakmai versenyzéseink során találkoztunk, ahol ezen moduláris gyártórendszerek mechanikai összeszerelésével, pneumatikus és villamos bekötésével, PLC programozásával, gyártási időre történő optimalizálásával foglalkoztunk. Volt versenyzőként szabadidőnkben a következő generációs versenyzőket oktadjuk, illetve trenírozzuk, így az ipari automatizálás eszközeivel és PLC programozással a mai napig kapcsolatban vagyunk.

Eközben megkezdjük egyetemi tanulmányainkat és szakirányként mindketten Automatizált gyártórendszerek modult választottuk, ahol leginkább beágyazott rendszerek tervezését, mikrokontrollerre, illetve FPGA-ra történő programfejlesztést tanuljuk.

Versenyzői, illetve már felkészítői tevékenységünk sok tekintetben megegyezik iskolai tanulmányainkkal, kivéve a vezérlő berendezéseket. Innen jött az ötlet, hogy ha már rendelkezésünkre állnak ipari gyártósori modulok, akkor azokból állítsunk össze egy technológia folyamatot és üzemeljük be beágyazott vezérlővel.

2. Célkitűzés

Célunk:

- egy 3 modulból álló gyártósori berendezés összeszerelése és annak beüzemelése ATMEL mikrokontrollert tartalmazó beágyazott vezérlővel,
- valamint a gyártási folyamat felügyeletét biztosítani egy személyi számítógépen futó alkalmazás segítségével.

További célunk, hogy munkánk befejeztével összehasonlítsuk a beágyazott rendszerrel, illetve PLC-vel történő beüzemelés folyamatát, idejét, illetve költségvetését és ezek alapján levonni a megfelelő tanulságokat.

3. Rendelkezésre álló eszközök

3.1 MPS Állomások

A FESTO MPS (Modular Production System) állomások a FESTO cég oktatási célra kifejlesztett eszközei. Olyan ipari elemekből állnak, amelyeket széles körben alkalmaznak gyártósori berendezésekben (pl.: pneumatikus szelepszigetek, munkahengerek, optikai szenzorok, villamos aktuátorok). Technológiai szempontból, az egyes modulok különböző feladatokat látnak el. Ezek összekapcsolásával a munkadarabokon általában adagolási, megmunkálási, szortírozási/palettázási folyamatokat hajtanak végre, így reprezentálva egy komplex gyártósori berendezés működését.

Az általunk választott MPS-ekből összeállított gyártósori rendszer, adagolási, összeszerelési, valamint palettázási művelet hajt végre.

3.1.1 Munkadarabok

Az MPS állomások által manipulált munkadarabok hengeres, 40 mm átmérőjű, 25 mm magas, 1,2 mm falvastagságú, üreges műanyag edények. Megkülönböztethetők szín, áttetszőség, anyag, illetve orientáció alapján.

Szín szerint lehetnek:

- piros
- fekete
- fém-színű

Áttetszőség szempontjából:

- teljesen átlátszó
- nem átlátszó

Anyag szerint:

- műanyag
- műanyag, fémezett festékkel

Orientáció szerint állhatnak:

- üreges oldallal lefelé
- üreges oldallal felfelé

Példánk során, azért csak fekete munkadarabokat használtunk, mert nem állt rendelkezésünkre olyan munkaállomás, amely szín szerint képes a munkadarabok adagolására, így a megrendelésnek megfelelő színű munkadarab kiadagolása nem lehetséges.



1. ábra Munkadarabok és kupakok

3.1.2 Kupakok

Rendelkezésünkre álltak még a munkadarabok lezárására alkalmas kupakok. Osztályozni anyag, valamint szín alapján tudjuk őket.

Szín szerint:

- piros
- fekete
- fém-színű
- szürke

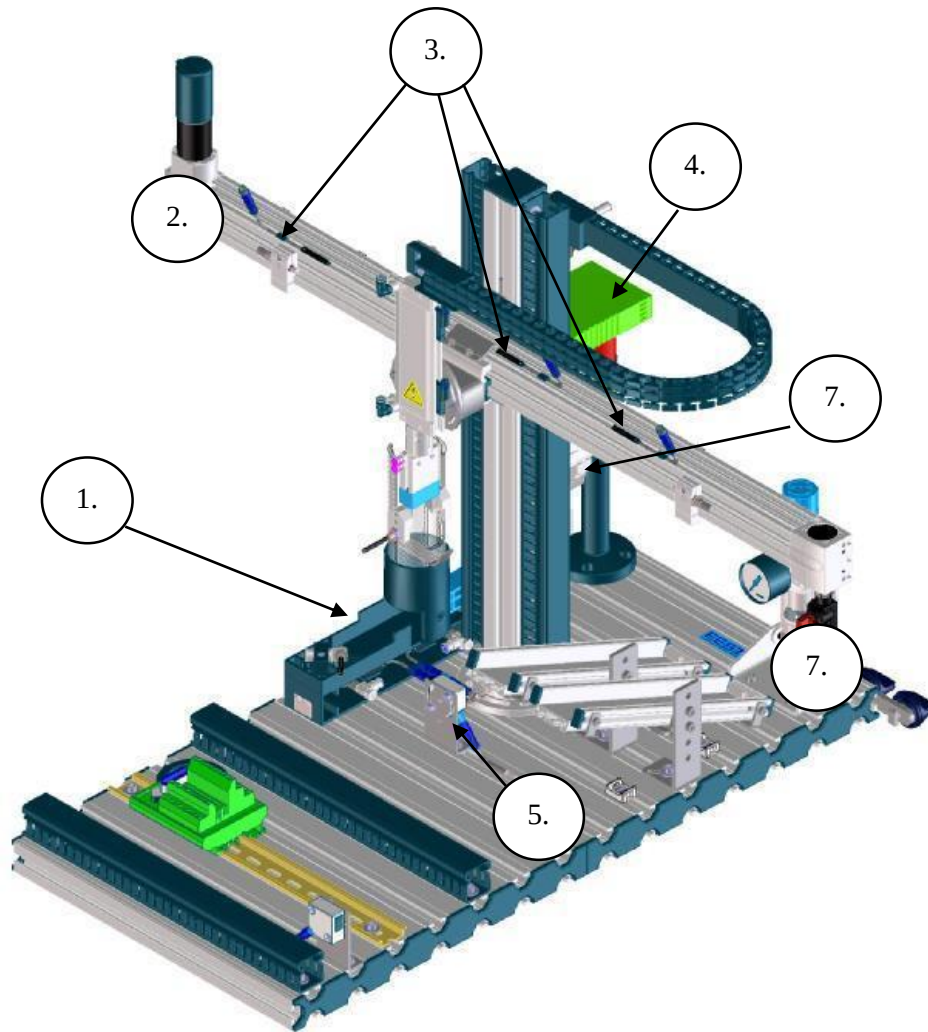
Anyag szerint:

- műanyag
- műanyag fémezett festékkel

Példánk során, csak műanyag, szín szerint, piros, fekete és szürke kupakokat használtunk fel.

3.1.3 Handling állomás

Fő feladata tulajdonképpen az ejtőtárban levő munkadarabok adagolása, valamint azok szállítása a következő állomásra, miközben képes arra, hogy megállapítsa a felvett munkadarab árnyalatát. Működtetéséhez 24V-os feszültségű 8 digitális be-, és 8 digitális kimenetre van szükség. A működés könnyebb megértése érdekében, az állomás egyes elemeinek működését, esetleg működési elvét ismertetnénk.



2. ábra Handling állomás CAD rajza

Főbb alkatrészek

1. Magazin

A munkadarabokat felülről lehet betölteni, maximum 5 darabot. Az adagolás egy kettősműködésű munkahenger segítségével történik, ilyenkor a munkadarab kikerül a hengeres tárolóból úgy, hogy azt felülről akadálymentesen megközelíthető.

2. Lineáris hajtás

Mozgató mechanikája egy fogas szíjhajtás, amire a lineáris hajtómű kocsija csatlakozik. Mozgatása az egység végén tengelykapcsolón keresztül, DC motor segítségével történik.

Az általunk használt rendszerben a hajtás kocsijára a pneumatikus megfogót vertikálisan mozgató munkahenger van felszerelve.



3. ábra EGC-TB típusú lineáris hajtás

3. Reed érzékelő

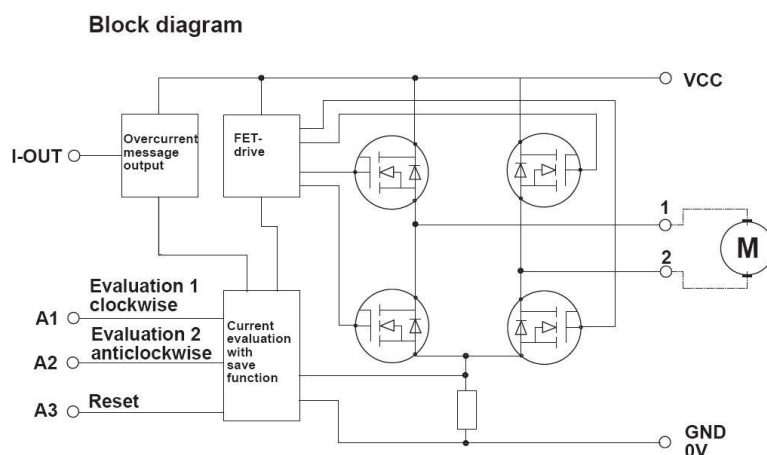
A magazin munkahengerén, illetve a lineáris hajtás hornyaiba Reed érzékelők vannak elhelyezve. Ezen szenzorok jeleit felhasználva tudjuk az illető aktuátor helyzetét meghatározni.



4. ábra Reed érzékelő

4. Motorvezérlő

Annak érdekében, hogy a DC motor árama ne a vezérlő egység kimentét terhelje, külön tápellátással rendelkező motorvezérlőt alkalmazunk. Mindkét irányba történő mozgást a vezérlőben levő H-híd teszi lehetővé. További funkciója még, hogy ha a motor túl nagy áramot vesz fel ütközés, vagy befeszülés következtében, akkor annak mozgását letiltja és egy kimeneti biten jelzi a hibát. Ezen áramkorlát beállítása potenciométer segítségével történik.



5. ábra Motorvezérlő blokkdiagramja

5. Tárgyreflexiós szenzor

Tárgyreflexiós optikai szenzor segítségével történik a munkadarabok jelenlétének vizsgálata. Amennyiben szenzor előtt lévő objektumról visszavert IR-fény elég intenzív, akkor a kiértékelő elektronika logikai magas szintet állít be az érzékelő kimenetére. Érzékenység beállítása egy potenciométer segítségével lehetséges a tokozás hátsó oldalán.



6. ábra Optikai kábel és a SOEG-L-Q30-NA-K-2L típusú tárgyreflexiós szenzor

6. Szelepsziget

Szelepsziget feladata, hogy az elektromos feszültségjelek hatására a benne lévő útszelepeket vezérelje. Az állomáson levő egységben 3 darab 5/2-es monostabil szelep található, amelyek a kilökö- és vertikális irányban mozgó munkahengert, valamint a pneumatikus megfogót működtetik.



7. ábra FESTO CPV-SC-MP típusú szelepsziget

7. Levegő előkészítő

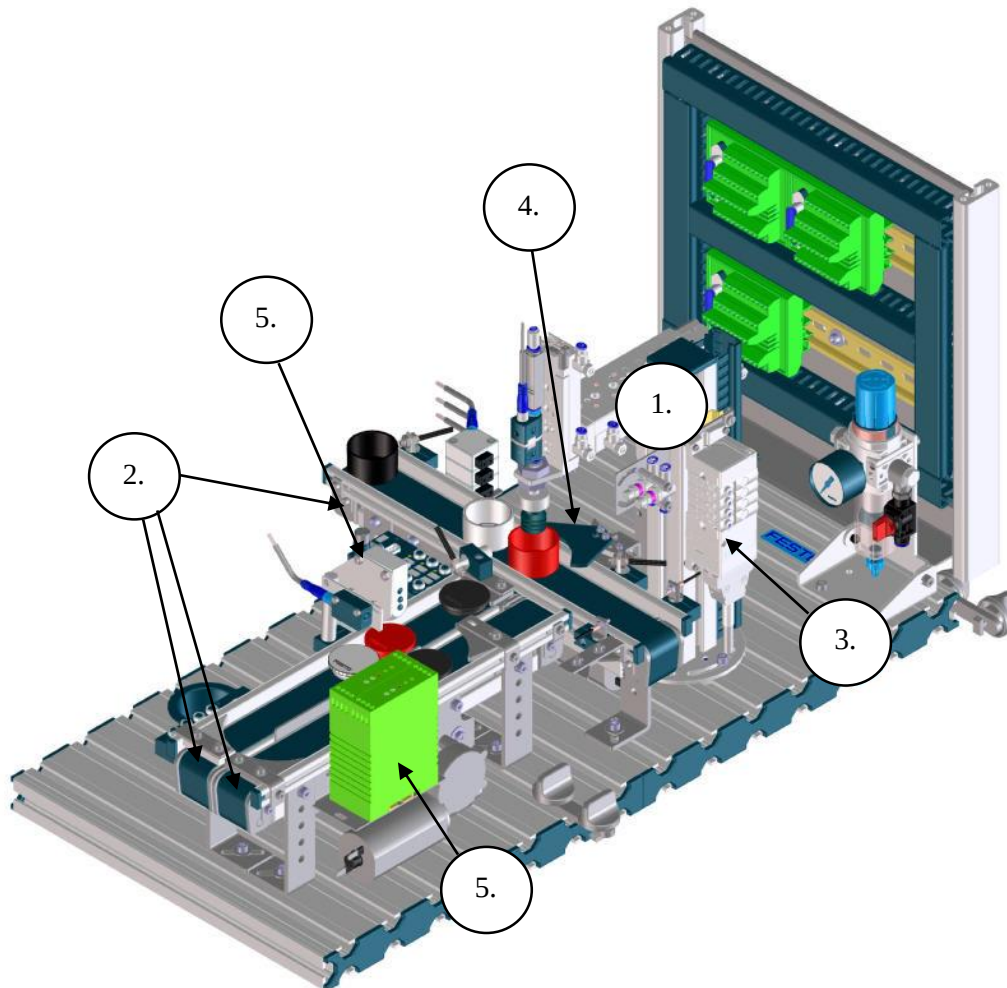
Levegő előkészítő feladata, hogy a bejövő sűrített levegő nyomását megfelelő szintre redukálja (ez a mi esetünkben 6 bar), valamint a víztartalmát minél alacsonyabb szinten tartsa.



8. ábra Festo levegő előkészítő

3.1.4 Cap állomás

Feladata a fő szalagra érkező munkadarab összeszerelési pozícióba szállítása, a megfelelő színű kupak kikeverése és ezek összeszerelése. Ezt követően az elkészült termék a következő állomásra juttatása. Működtetéséhez 24V-os feszültségű 13 digitális be-, és 9 digitális kimenetre van szükség.

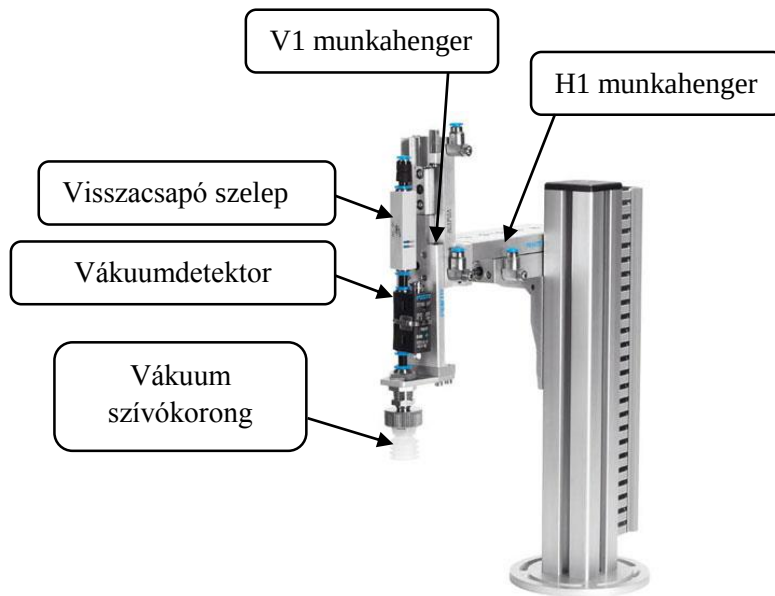


9. ábra Cap állomás cad rajza

Főbb alkatrészek:

1. Pick&Place modul

Egy horizontális, valamint egy vertikális irányban mozgó munkahengerből áll, valamint az utóbbihoz csatlakoztatott vákuum szívókorongból, amely a kupakok megfogásához szükséges. Munkahengerek véghelyzetéről szintén Reed érzékelők adnak jelet.



10. ábra Pick&Place modul

2. Szállítószalag

Működtetése 24V-os DC motor segítségével történik. Cap állomás összesen három darabot tartalmaz. Az első, azaz fő szállítószalag feladata a munkadarabok áramoltatása, a maradék kettő rendeltetése a kupakok keringetése óra járásával ellentétes irányban, a felszerelt terelőlapátok segítségével.



11. ábra Szállítószalag

3. Szelepsziget

A Pick&Place modulhoz tartozó munkahengerek, valamint a fordítómű vezérléséhez szükséges útszelepeket és a vákuum kialakításához szükséges vákuum ejektort tartalmazza.

4. Fordítómű

Feladata az érkező munkadarabok pontos pozícióban történő megállítása, hogy az összeszerelési metódus sikeresen végbemehessen.

5. Tanítható színérzékelő szenzor

Összesen háromféle szín megkülönböztetésére, 3 darab 24V-os feszültségű kimenettel rendelkezik. A kívánt szín tetszőlegesen betanítható a tokozáson elhelyezett nyomógombok (S – Set, T – Teach) segítségével. Beállításánál az érzékelő számára egy mintát kell biztosítani, kiválasztani a megfelelő kimeneti csatornát, valamint meghatározni az érzékelési toleranciát.



12. ábra SOEC-RT-Q50-PS-S-7L típusú színszenzor

6. Motorvezérlő

Handling állomásnál már ismertettük működését. Feladata a két kupakokat keringető szállítószalag vezérlése mindkét irányba.

.

3.1.5 Robotállomás

Az ipar széles körben alkalmaz robotokat, mind anyagmozgatási mind pedig szerelési műveletek elvégzésére.

A legegyszerűbb felépítésűek az úgynevezett **manipulátorok**. Ezek 3 transzlációs tengelyt tartalmaznak. Elterjedt még a megfogó vagy a szerszám további rotációs tengelyre történő szerelése, amivel megvalósítható a munkadarab forgatása is. Csuklóképletük TTT vagy TTTR, szabadságfoka pedig 3+1. Az anyagmozgatás a legelterjedtebb felhasználása. A portálok felépítésének köszönhetően nagy terhek mozgására is alkalmasak.

Delta robotok a felépítésükről kapták a nevüket. Ezek zárt láncú, párhuzamos kinematikájú robotok, azaz nem valósítható meg mozgás csak az egyik tengelyt használva. A 3 felfüggesztési pontban tartalmaznak egy-egy rotációs tengelyt. A középpontjában rögzített, egymással 120°-os szöget bezáró elrendezés biztosítja a nagy merevségét és a pontos pozicionálást valósítja meg. A munkavégzés tartományát a szervók elfordulási szöge és merevkarok hossza határozza meg. Elsősorban Pick&Place feladatokat lát el, mivel függőleges irányban minimális elmozdulásra képesek. Felépítésének köszönhetően nagy sebesség mellett is jól pozicionálható.

A **SCARA** robotok szintén nagy sebességű Pick&Place műveletek megvalósítására alkalmasak. Felépítését tekintve legalább 2 rotációs tengellyel rendelkeznek, de ezekkel csak x, y irányú mozgás valósítható meg. A z tengely menti mozgást az utolsó, transzlációs tengely biztosítja. Előnye a nagy sebessége viszont függőleges irányban minimálisan terhelhető és kicsi függőleges irányban a munkatere.

A **Humanoid** robotok a legszélesebb körben felhasználható robotok a gyártási folyamatokban. Felépítését tekintve nyíltláncú, elágazás nélküli merev robot. Az egyes szegmenseket összekötő csuklók (joint) adják meg a robot szabadságfokát. Csuklóképletüket tekintve RRRRRR vagy RRRRRR a legelterjedtebb. Léteznek 6-nál több csuklóval rendelkező robotkarok is. Ennek előnye, hogy redundánssá teszi a mozgást, azaz egy pozíció több féle képen is megközelíthetővé válik. Felhasználásuk igen széleskörű, és ez egyben meghatározza felépítésüket is. Például egy ívhegesztő robotnak nagyobb pontosságot kell biztosítania, mint ha csak egy anyagmozgatási feladatot látna el. Vagy egy ponthegesztő szerszám mozgásához jóval robosztusabb felépítés szükséges, mint egy festőrobotnál. Néhány felhasználásuk: ívhegesztés, festés, ponthegesztés, szerelés, anyagmozgatás, palettázás, ragasztás.

A robotok pozícióinak számításához szükséges valamilyen koordináta rendszert megadni. A legtöbb humanoid robot több féle koordináta rendszert is használ, az itt felsoroltak mindegyike

Descartes-féle. Minden robotnak meg kell adni egy világ koordináta rendszert, mely alapján értelmezi az egyes pozícióit. Gyárilag ez megegyezik a robot alap koordinátarendszerével, de eltolható a munkatér bármely pontjára. Későbbiekben a pozíciók meghatározása ehhez viszonyítva történik. Ezeken kívül használunk még egy TOOL koordináta rendszert is. Ennek középpontja a TCP(Tool Center Point), amely alapesetben a robot 6. tengelyének középpontjában helyezkedik el. A TCP-t a robot szerszámtól függően, elhelyezhetjük annak valamely pontjában. (például az ívhegesztő hegyére, vagy festékszóró szórófejére). Így megvalósítható a szerszám egy adott pályán történő mozgatása. Természetesen a robot vezérlő ezeket a koordinátákat is a világ koordinátarendszerhez viszonyítja. A szerszám orientációja is megadható, ez koordináta rendszertől függetlenül mindig a TCP ponthoz képest történik. Az X,Y,Z tengelyekhez tartozó orientációk α, β, γ (az általunk használt fejlesztőkörnyezet A,B,C jelölést használja, későbbiekben mi is így hivatkozunk rájuk) az egyes tengelyek menti elfordítást adja meg.

Az általunk összeállított gyártási folyamat utolsó szegmense egy Mitsubishi Rv-2AJ típusú ipari robotkar, amely a munkadarabok palettázását végzi. Ez egy 5 tengelyes humanoid robot. Az 5 tengelyből adódóan 5 szabadságfoka van, a csuklóképlete pedig RRRRR. Maximum teherbírása 2kg, mely az általunk használt munkadarabhoz és megfogó szerszámhoz tökéletesen megfelel. Megfogó szerkezetnek egy FESTO HGP-16-A-B típusú pneumatikus párhuzamos megfogót használunk, a munkadarabhoz illeszkedő megfogó pofákkal.-

A robot állomás felépítése

A munkadarab az előző állomáson elhelyezett szállítoszalagról egy fészekbe érkezik, mely biztosítja a robot számára a pontos pozicionálást.



13. ábra Fénykép a robotállomásról

Innen a robot az elkészült termékeket, egy palettára helyezi. A biztonság érdekében, egy-egy tárgyreflexiós optikai szenzorral figyelhető, hogy érkezett-e munkadarab és hogy van-e paletta az állomáson. Ezek mindegyike a robotvezérlő digitális I/O portjára köthető.

3.2 Mikrokontroller board

Az állomások megvizsgálása után, már csak a célnak megfelelő processzor kiválasztása maradt. A művelet nem igényel nagy számítási sebességet, elegendő egy 8 bites mikrokontroller a feladat megvalósítására. Tanulmányaink során Atmel vezérlőkkel foglalkoztunk, éppen ezért AVR-t választottunk a feladatra. Rendelkezésünk állt néhány AT90CAN128-ra épülő demo board, amelyek már csak a beépített CAN interfészt tekintve is, tökéletesen megfelelnek a feladatainkra.

A panelon található 5 darab nyomógomb és 8 darab LED. Ezeknek jelentősége leginkább csak az egyes modulok tesztelési fázisában volt. Továbbá egy FTDI FT232-es UART-USB illesztő IC és hozzá egy USB-B csatlakozó. A panelon még kivezetésre került az SPI lábak csatlakozása. Ez későbbiekben könnyedén felhasználható SPI-on keresztül csatlakozó perifériákhoz. Illetve a JTAG kivezetéseket használtuk a mikrokontroller felprogramozására, egy JTAG-ICE programozó segítségével. A panel tartalmazza a CAN busz meghajtásához szükséges áramkört is.

Jelen dolgozat keretein belül csak az AVR általunk is használt interfészeit mutatjuk be. Mint korábban is említettük ez egy 8 bites mikrokontroller. Maga a processzor RISC architektúrára épül és akár 16MIIPS (**M**illion **I**nteger **I**nstructions **P**er **S**ec) teljesítményre is képes 16MHz-es órajel esetén. Ezen kívül tartalmaz egy hardware-es szorzó áramkört is. A belső flash memóriája pedig 128Kbyte. 6 db, a felhasználó által konfigurálható, 8 bites port és egy darab 5 bites valósítja meg a környezetével a kapcsolatot.

Az SPI (**S**erial **P**eripheral **I**nterface) egy full-duplex adatátvitelt 3 vezetéken megvalósító szinkron kommunikáció. Ez a 3 vezeték a MISO (Master In Slave Out), a MOSI (Master Out Slave In) és az SCK. Az SCK órajel biztosítja a busz számára a szinkron órajelet. Az AVR működhet mind master mind pedig slave eszközként is. Jelen esetben Master-ként használtuk, ezért szükséges még egy negyedik, SS (Slave Select) vonalat is alkalmazni. Az SPI megvalósítása lényegében egy shift regiszter, ami a kommunikáció során a regiszter MSB oldalán kishifteli a benne tárolt adatot a MOSI vonalra. Eközben a MISO vonalon érkező adatok beíródnak a shift regiszter LSB felőli oldalán. Így valósítva meg a full-duplex kommunikációt. Ha slave eszközként is shift regisztereket alkalmazunk, akkor igen egyszerűen megvalósíthatunk egy port bővítő áramkört. Inicializáláskor az AVR-ben elég megadnunk az SCK frekvenciáját, azt hogy master-ként vagy slave-ként kívánjuk használni, hogy az MSB vagy az LSB felőli oldalról kezdje a küldést, illetve hogy megszakítást akarunk-e használni az adatátvitel befejezésekor.

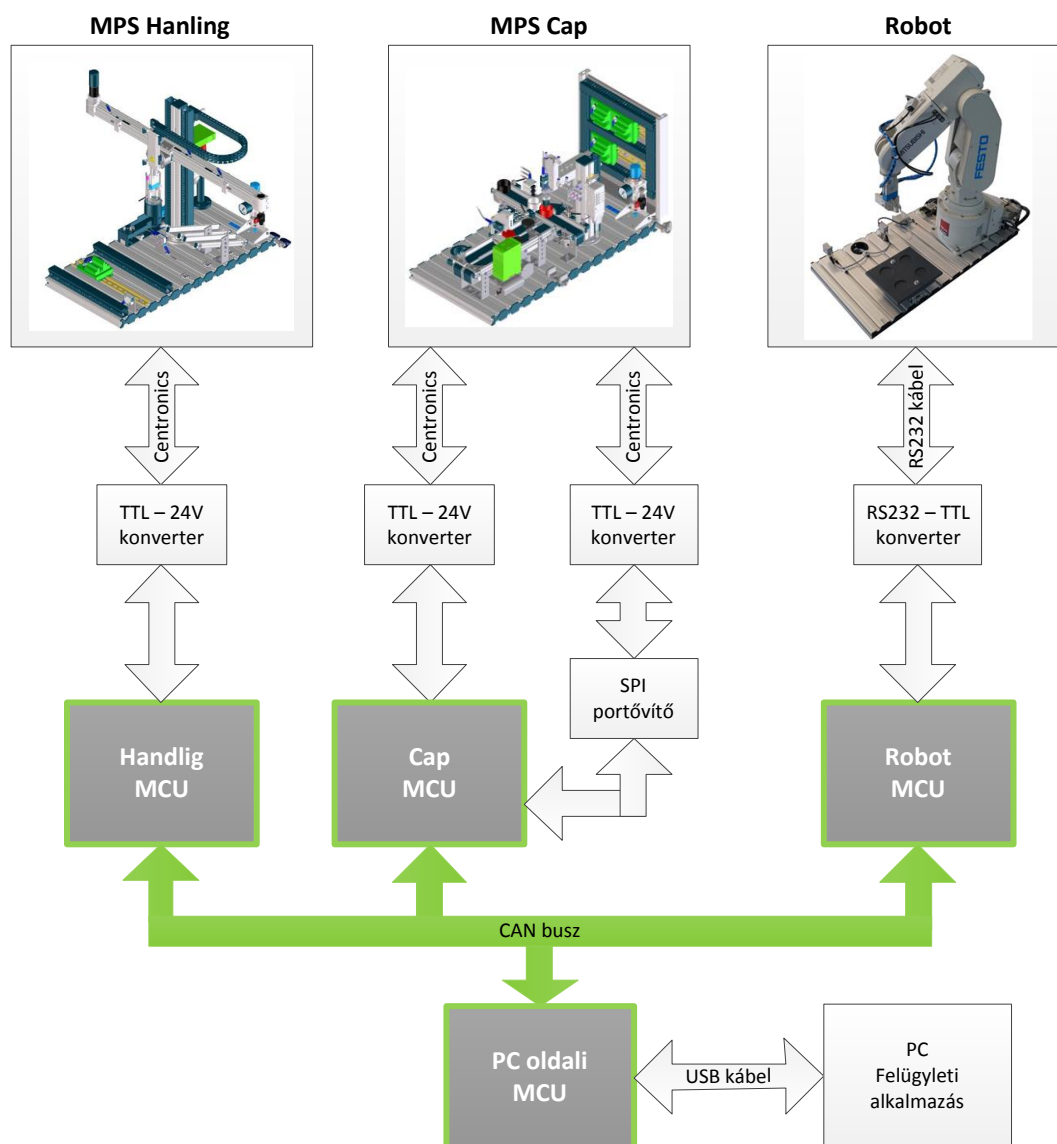
A mikrokontrollerrel megvalósítható további soros kommunikáció az USART0, USART1 (**U**niversal **S**ynchronous and **A**ynchronous serial **R**eceiver and **T**ransmitter) segítségével. Ezek segítségével valósítottuk meg a PC és a robot felőli kommunikációt. Ahogy a nevéből is látszik, mindegyik működhet szinkron és aszinkron módban is. Ezen kívül további beállítási lehetőségei az adatátvitel sebessége, az adatbitek száma, a stop bitek száma és a paritás bit. Illetve engedélyezhető külön-külön a küldő (Tx) és fogadó (Rx) ág, valamint a hozzájuk tartozó megszakítások. Mi aszinkron üzemben használtuk, a következő beállításokkal: Baud Rate = 9600, 8 adatbit, 2 stopbit és páros paritásbit. Ezen kívül az érkező keretek detektálására a Receiver Interrupt-ot használjuk.

A kicsit is összetettebb feladatok megvalósításához általában elengedhetetlenek a Timerek használata. Ezek segítségével adott időközönként hajthatunk végre feladatokat. Mi felhasználásunkban erre egyik legjobb példa a shift regiszteres port bővítő lekérdezése, a Timert kiszolgáló megszakítási rutinból. Az AT90CAN128 2db 8 bites és 2db 16 bites Timert tartalmaz. Ezek legfontosabb beállítási lehetőségei, hogy milyen üzemmódban működjenek, mekkora frekvenciával, és hogy mi történjen ha elszámolt a megadott értékig (megszakítás, pwm láb,

stb...). Az SPI-os példához használt beállításaink a következők: CTC (Clear Timer on Compare Match) üzemmód. Azaz a Timer 0-tól elkezd számolni, és amikor eléri a beállított komparálási szintet, akkor nullázódik a számláló és kezdődik előlről. 1ms-os megszakítást kívántunk létrehozni, 16MHz-es órajel mellett, ehhez a komparálási szintet (OCR0A regiszter) 124-re kellett állítani, az előosztást pedig 64-re.

4. Megvalósított folyamat

Az általunk összeállított és felprogramozott gyártósori berendezés felépítését és annak működtetését az alábbiakban ismertetjük.

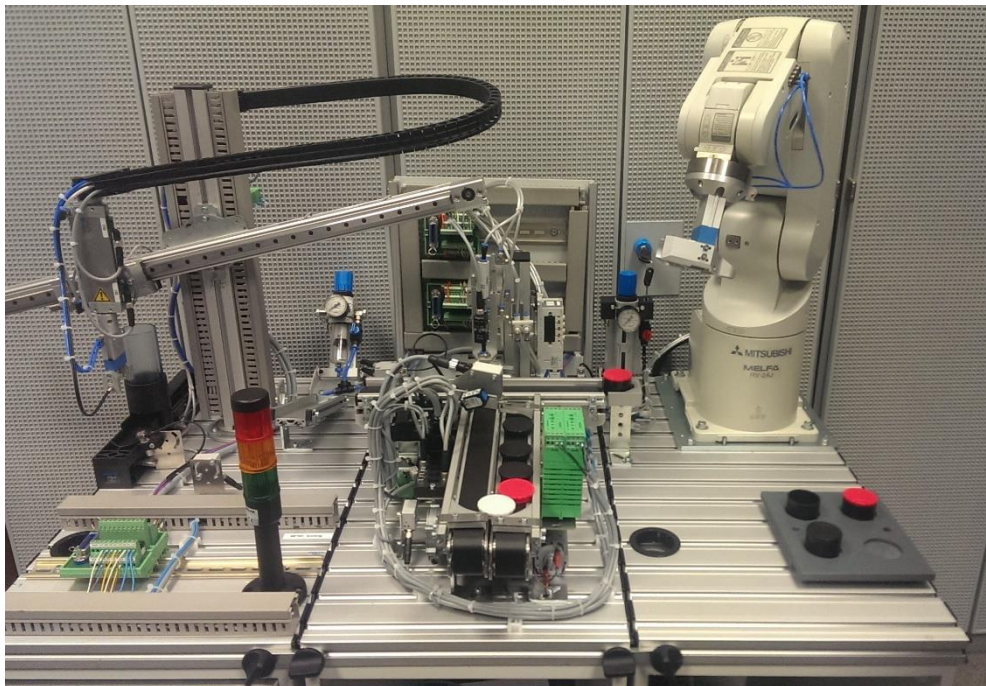


14. ábra A teljes rendszer blokkvázlata

4.1 A rendszer kialakításának ismertetése

A fenti ábra a gyártósori modell felépítését reprezentálja:

- Handling állomás 8-8 digitális I/O-val működtethető, ezért a mikrokontrollerhez történő csatlakoztatására 1 darab TTL – 24V szintillesztő szükséges.
- Cap állomás 13 digitális be-, illetve 9 digitális kimenet által működtethető, éppen ezért az általunk választott mikrokontroller kevés szabadon maradt I/O portja miatt egy SPI port bővítő modulra volt szükségünk. Így az egyik szintillesztő a mikrokontrollerhez közvetlenül, a másik pedig a port bővítőn keresztül csatlakozik.
- Robot állomás a hozzá tartozó mikrokontrollerrel soros porton kommunikál. Mivel az általunk használt vezérlő panelje alpból nem rendelkezik sorosporti csatlakozóval, ezért RS232 – TTL illesztőre volt szükségünk.
- A PC oldali mikrokontrollerünk a felügyeleti PC-vel USB-n keresztül csatlakozik, amely kommunikációs interfész már ki volt alakítva a panelen.
- Az egyes mikrokontrollerek CAN buszra vannak csatlakoztatva, amely buszon keresztül kommunikálni egymással.



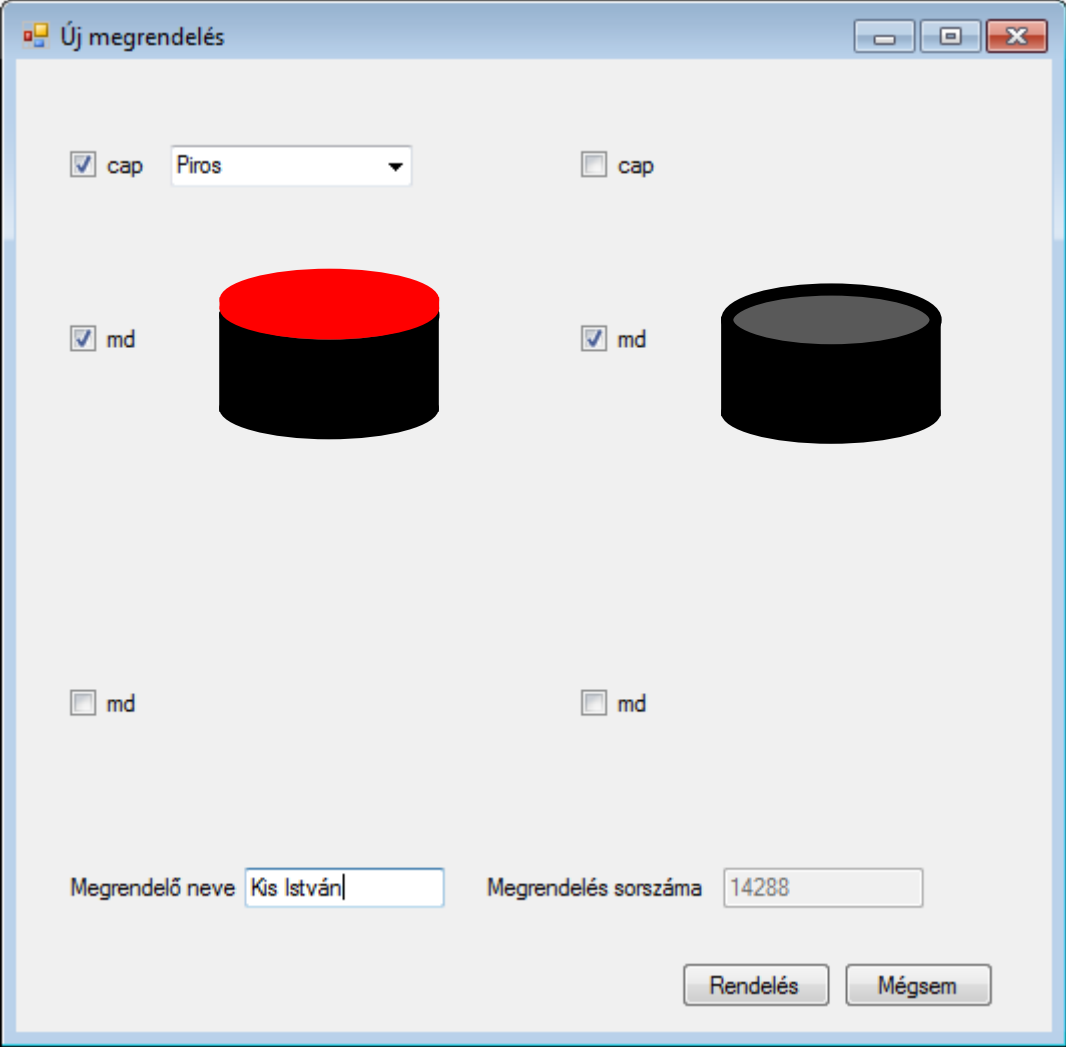
15. ábra Fénykép az összeállított rendszerről

4.2 A működtetés rövid leírása

Megrendelési folyamat

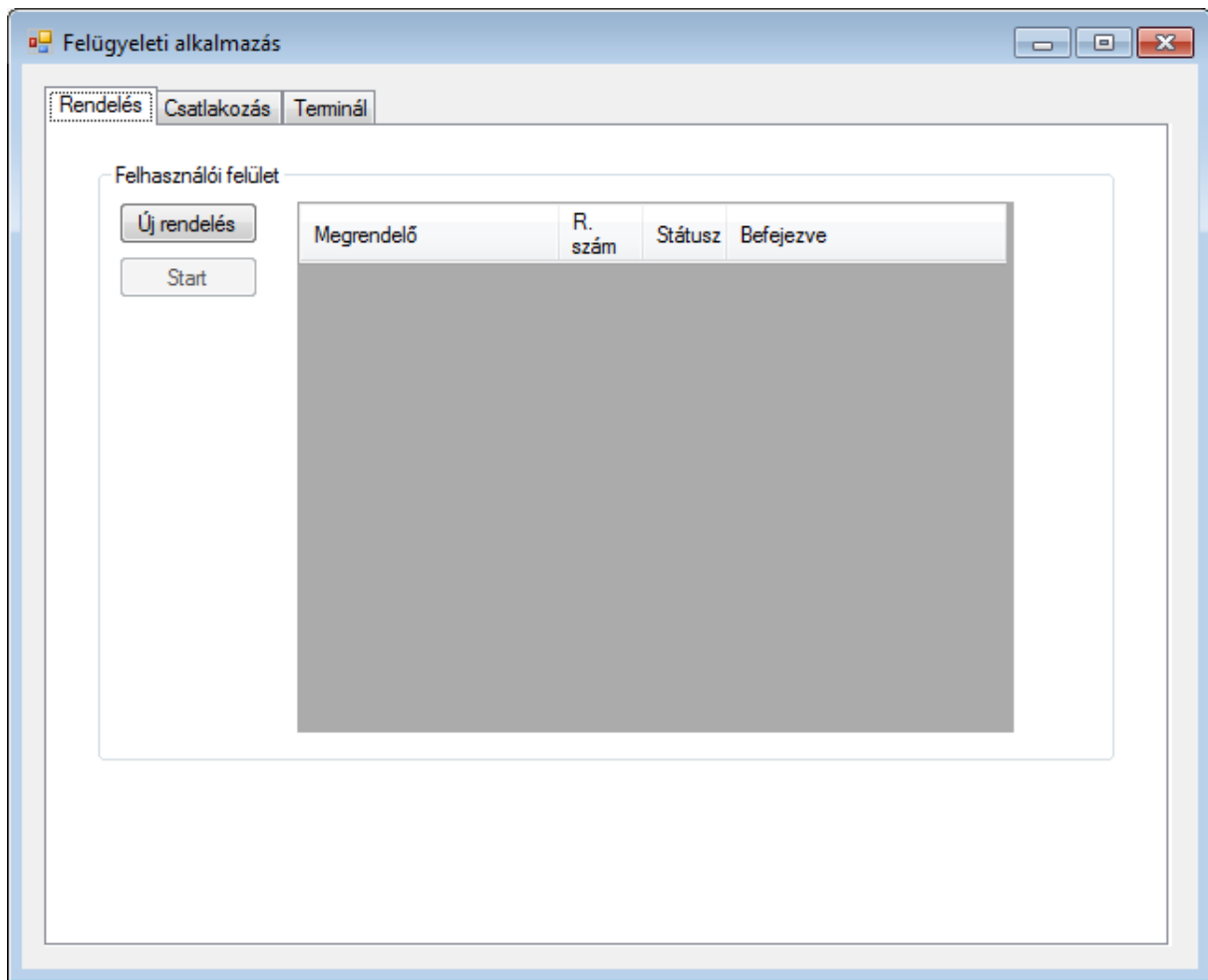
A személyi számítógépen futó alkalmazás segítségével tudunk rendelési művelet végrehajtani. Az új rendelés nyomógombra kattintva felugrik egy ablak, ahol a rendelésünket vehetjük fel.

Maximum négy darab fekete munkadarabot kérhetünk egyszerre, de lehetőségünk nyílik kevesebb megrendelésére is. További opcióként kérhetünk az munkadarabokra kupakot, amely kupakok színe piros, fekete, vagy szürke lehet.



16. ábra A PC-n futó megrendelőablak

Ezt követően meg kell adni a megrendelő nevét, majd a megrendelés gombra kattintva a program regisztrálja az adatokat. Ekkor visszatérünk a fő programablakunkba, ahol egy táblázatban feltüntetésre kerül az újonnan felvett rendelés. Start nyomógomb segítségével az alkalmazás elindítja a gyártási folyamatot.



17. ábra A felügyeleti program

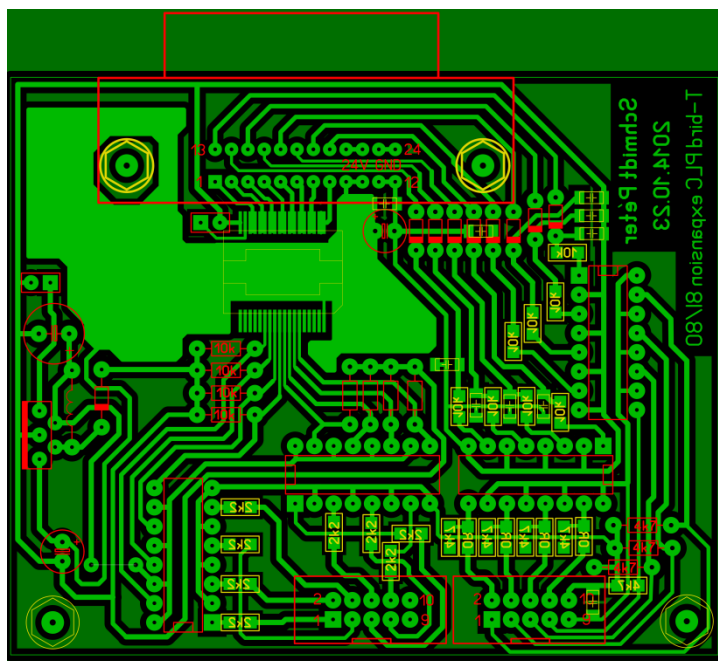
Gyártási folyamat

A Handling-en lévő magazin adagol egy munkadarabot, amit egy pneumatikus megfogó segítségével a következő állomásra továbbít. Ha rendelés során nem kért kupakot a megrendelő, akkor a Cap állomás a szállítószalagon a robot számára továbbítja a gyártmányt. Ellenkező esetben a munkadarab összeszerelési pozícióba kerül, majd a berendezés kikeresi a megrendelésnek megfelelő színű kupakot és elvégzi az összeszerelést. Ezután a futószalag továbbítja a következő állomás felé a készterméket, ahol a robot elvégzi a palettázási műveletet.

Ha megrendelő csak egy munkadarabot rendelt, akkor a gyártási procedúra be is fejeződött, és a számítógépes alkalmazáson megjelenik a befejezés pontos dátuma és időpontja. Innen tudjuk, hogy az adott termékvariáció elkészült. Több munkadarab esetén értelemszerűen a Handling állomás újra adagol egy munkadarabot és folytatódik a gyártás.

5. Tervezés és fejlesztés

5.1 TTL – 24V konverter



18. ábra TTL - 24V NYÁK terve

Az MPS állomások az iparban elterjedt 24V-os tápfeszültségről üzemelnek, az AT90CAN128 mikrokontroller azonban 5V-os TTL jelszintet használ, éppen ezért szükség volt egy szintillesztő áramkörre, hogy a két eszköz kompatibilis legyen egymással. Szintillesztésen túl további feladata még a galvanikus leválasztás megvalósítása, az adatvonalakon fellépő zavarok szűrése és a megfelelő csatlakozófelület kialakítása.

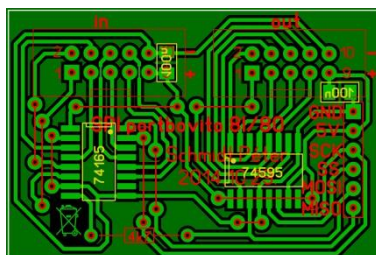
A mikrokontroller felőli oldalon 2db 10 pin-es szalagkábel csatlakozást használunk, amelyen keresztül az áramkör megkapja az 5V-os tápfeszültséget is. A gyártórendszerrel egy 24pin-es Centronix csatlakozón keresztül valósul meg a kapcsolat, ahonnan a 24V-os tápellátás is megtörténik. Tehát külön tápcsatlakozóra nincs szükség. Ezen az oldalon a bemeneteket elláttuk egy-egy soros diódával, ezzel megvalósítva a fordított polaritás elleni védelmet. Egy 100nF-os kondenzátor segítségével szűrjük a bemenetekre érkező impulzus jellegű zajokat. Ezután egy ISP521-4 négy csatornás optocsatoló következik, ami megvalósítja a galvanikus leválasztást (a kimenete és a bemenete közti átütési feszültség $5,3V_{RMS}$ és $7,5kV_{PK}$) és a szintillesztést is egyben. Kikapcsolt állapotban lehúzó ellenállások biztosítják az AVR bemenetén a logikai alacsony szintet.

A kimeneteken az STMicroelectronics VN808 típusú IC-jét alkalmaztuk. Ez egy 8 csatornás nagyszintű meghajtó áramkört tartalmaz. (Az N csatornás MOSFET kimenetnek köszönhetően minimális a rajta eső veszteség bekapcsolt állapotban. $R_{DS}=150\text{m}\Omega$, tehát 0,5A kimeneti áramnál is csak 37mW. Névleges feszültsége $V_{CC}=45\text{V}$, így a 24V-os rendszerben tökéletesen alkalmazható.) Az áramkörben beépített túláramvédelem, hőmegfutás elleni védelem és alacsony tápfeszültség elleni védelem van. A standby áramfelvétele minimális, ami energiatakarékossági szempontból fontos. Az egyes csatornák kapcsolási ideje $100\mu\text{s}$ alatti (katalógus adat: $t_{ON}=50\mu\text{s}$, $t_{OFF}=75\mu\text{s}$), ami egy gyártó berendezéshez tökéletesen megfelel. A tokból ki van vezetve egy open source-os STATUS láb is, melyen a védelem állapota akár mikrokontrollerrel is feldolgozható hiba detektálás céljából. Kimeneti áram korlátozva van, a katalógus alapján ez 0,7-1,2 A-re maximalizálható. Méréseink alapján $I_{max}=0,8\text{A}$. Ez elegendő a gyártórendszerekben alkalmazott szelepek, relék működtetéséhez. Az IC bemeneteinek meghajtásához legalább a $V_{CC}/2$ feszültségre lenne szükség. (Jelen esetben ez 12V, de a méréseink alapján ez nem elegendő. 15V szükséges a biztonságos üzemeltetéshez.). A mikrokontroller kimenete és a VN808 bemenete között szintén az ISP521-4 optocsatolót alkalmazzuk, melynek a fő feladata itt is a galvanikus leválasztás és a TTL-15V közti szintillesztés. Az optotranzisztor I_C kimeneti árama 50mA, ami elegendő a kapcsoló áramkör meghajtására.

A szükséges 15V-ot egy LM1575-15 step-down típusú stabilizátor állítja elő a 24V-os tápfeszültségből. Azért választottunk kapcsolóüzemű stabilizátort, mert minimális a vesztesége, kevesebb a disszipált teljesítménye és így nem szükséges hozzá külön hűtőfelület kialakítása.

Az áramkör megvalósítása egyoldalas nyomtatott huzalozású kártyán történt, nagyrészt felületszerelt alkatrészekből. A VN808 PowerSO tokozású, így a hűtőfelület kialakítását a NYÁK-on kellett megvalósítani. Az alsó rézréteg esetében 6cm^2 hűtő felülettel már elérhető a 15°C/W -os disszipáció. (A felület további növelésével ez már csak minimálisan javítható.) Az általunk kialakított hűtőfelület területe körülbelül $6\text{-}7\text{cm}^2$. Az optocsatolók DIP16-os tokozásúak(furatszerelt) és foglalatba kerültek, így egy esetleges zárlat után egyszerűen elvégezhető a cseréjük.

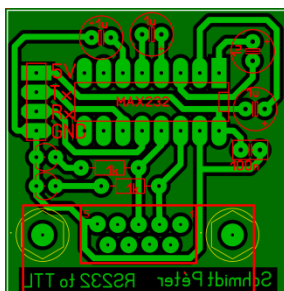
5.2 SPI port bővítő



19. ábra SPI portbővítő NYÁK terve

Mivel a második állomás működtetéséhez 9 darab kimenetre és 13 darab bemenetre van szükség, de az általunk használt AT90CAN128 panelen csak két port áll rendelkezésre, ezért szükségeltett valamilyen port bővítő megoldás. Az egyszerűsége és a kis áthidalandó távolság (néhány cm) miatt esett a választásunk az SPI-os port bővítő megoldásra. Másik nagy előnye, hogy szabványos kommunikáció és a mikrovezérlők többségében megtalálható. Így széleskörűen alkalmazható minden SPI interfésszel rendelkező mikrokontroller perifériájaként gyártó és típustól függetlenül. Kimenetek vezérléséhez 74HC595-re, a bemenetek kezeléséhez pedig a 74HC165-re esett a választásunk. Mindegyik áramkör CMOS felépítésű, az AVR által használt 5V-os jelszint megfelelő hozzá. A 74165 egy 8 bites párhuzamos bemenettel és soros kimenettel rendelkező shift regiszter. A Q7-es soros kimeneti lábát a busz MISO vonalára kötve valósítható meg a soros adatvonal az AVR felé. Amíg a PL lába alacsony szinten van, folyamatosan olvassa be a 8 bites adatot. Majd amint logikai egyre vált, a tárolóiból minden egyes órajelre kiléptet egy bitet. Kommunikáció befejezte után, a PL láb újra alacsony szintre kerül és folytatja a párhuzamos busz beolvasását. A 74HC595 pedig egy 8 bites soros bemenettel és párhuzamos 3-state kimenettel rendelkező shift regiszter. Található benne egy storage regiszter és egy shift regiszter is. Ennek köszönhetően az adatbeolvasás ideje alatt is megtartják a kimenetek a storage regiszterben tárolt értéküket. A DS lábát az SPI busz MOSI lábára kötve fogadja a soros adatot, amint az ST_CP lába alacsony szintre kerül. Ezt a kommunikációban az SS vonal valósítja meg. Mivel az adat beolvasásához a 74165-nek magas szintű engedélyezés kell, a 74595-nek pedig alacsony szintű, ezért szükséges egy invertert az SS vonalra beiktatni. Ez egyszerűen megvalósítható egy tranzisztorból.

5.3 TTL-RS232 illesztő



20. ábra TTL - RS232 NYÁK terve

Az RV-2AJ robotkar vezérlője RS232-es kommunikációs interfészt, az AT90CAN128 aszinkron soros kommunikációja (UART) pedig TTL jelszintet használ, ezért szükséges volt valamilyen illesztő fokozat beiktatása. Az erre a célra gyártott ICL232 IC-t választottuk, amely előállítja az RS232-nek szükséges $\pm 10V$ feszültséget és el is végzi a konverziót. Az Rx, Tx lábakon kívül be kellett kötni az RTS, CTS lábakat is, különben a robot nem tudta megnyitni az RS232 COM portot. Ezeket a NYÁK-on egyszerűen csak rövidre kellett zárni.

5.4 Kommunikációk

A rendelkezésre álló eszközökből és a megvalósítandó folyamatból adódik, hogy minden állomás képes lesz önállóan ellátni a feladatát. De az egyes állomások között szükség lesz valamilyen kommunikációra. Még mielőtt nekiálltunk volna a tényleges program fejlesztésének, először is meg kellett határoznunk a teljes rendszerre, hogy ki, mikor, kinek küldjön üzenetet és mit tartalmazzon az az üzenet. Először tekintsük át, hogy a gyártósor egyes állomásai között milyen üzenetekre is lesz szükség. Mindegyiknek szüksége lesz egy INIT parancsra, ami hatására felveszik az alaphelyzetet. Ez akár ugyan az az üzenet is lehet minden állomásnak. A Handling station várni fog egy HS_START parancsot, ami hatására elkezdi egy munkadarab adagolását. Valahogy tudatnia kell a következő állomással, illetve a PC-n futó adatbázissal, hogy befejezte az adott műveletet. Ezért szükség lesz egy HS_MC motion completed üzenetre is. A rendszerhez használunk egy jelzőoszlopot is, a gép állapotának visszajelzésére. Mivel ez is a Handling Station-ön kapott helyet, ezért a további egységek üzemállapotáról is kell információt fogadnia. Például inicializálásnál, amíg nem vette fel minden állomás az alaphelyzetét, addig a piros lámpa villog. A Cap Station is hasonlóan a Handlinghez vár egy indító jelet, és a folyamat végén visszaad egy CAP_MC motion completed jelzést. Azonban itt nem elegendő egy indító jelet adni neki, ugyanis szükséges megadni a

megrendelésnek megfelelően, hogy az adott termékre milyen színű kupak kerüljön. Ezért itt nem is szükséges külön indítójel, hanem indulhat a folyamat amint megkapta a kupak színét. A Robot Station pedig annyit vár, hogy a paletta melyik helyére helyezze a gyártócellából kikerült terméket (ROB_POS1...ROB_POS4). Ez a Cap Station-höz hasonlóan egyben az indító jele is. Azonban a megfogási magasság miatt itt is számít, hogy van-e kupak a munkadarabon. A színe már lényegtelen ezért csak a kupak megléte a kérdéses (ROB_CAP, ROB_NCAP). Illetve nem maradhat el a ROB_MC folyamat vége jelzés sem.

Következő lépés a megfelelő kommunikációs interfész és protokoll kiválasztása. Az összeszedett üzenetekből látható, hogy az egyes állomások kommunikálnak egymás felé és a PC felé is. Tehát csak egy pont-pont kapcsolat a PC felé vagy csak a gyártósor elemei között nem elegendő. Minden egyes üzenethez megvalósítani egy pont-pont kapcsolatot pedig nem lenne gazdaságos és későbbiekben megnehezítené az esetleges bővítést is. Ezek alapján arra jutottunk, hogy valamilyen busz struktúrára lesz szükségünk. Ezután végignéztük milyen buszrendszerek alkalmasak ipari környezetben is. Az iparban a legelterjedtebb buszok RS485-re vagy CAN-re épülnek. Illetve nem busz struktúrájú, de széles körben alkalmaznak még Ethernet hálózatot is. Ehhez szükség lett volna külön perifériára a mikrokontroller mellé, míg az RS485-höz és a CAN-hez csak egy buszmeghajtóra van szükségünk, ezért az Ethernet ötletét el is vetettük. Egyes üzeneteket mindenkinek meg kell kapnia, valamelyik pedig csak néhány állomásnak, vagy csak az egyik állomás számára hordoz információt. Ennek megvalósítására és a megfelelő hibadetektálásra külön protokollt kellett volna kidolgoznunk RS485-re. Ezzel szemben a CAN üzenetorientált, és számos módszert alkalmaz hibadetektálásra, így lényegesen megrövidíti a tervezés idejét. Ráadásul a választott mikrokontroller tartalmaz is egy beépített CAN interfészt.

Azonban nem valósítható meg minden kommunikáció CAN buszon keresztül, hiszen a PC és a robot nem rendelkeznek CAN interfésszel. Ezért mindegyik mellé került egy mikrokontrolleres vezérlő, ami értelmezi az üzeneteket és továbbítja azokat a megfelelő formában. Az RV-2AJ roboton csak RS232 kommunikációs lehetőség van. Ehhez az AVR UART-ját használtuk fel. Azonban, hogy „melyik pozícióba mozogjon”, és hogy „van-e kupak” információt már egy stringbe egyesítve kapja meg a különálló üzenetek helyett.

PC oldalon pedig egy FTDI chip segítségével oldottuk meg az AVR UART – USB kommunikációt. A mikrokontroller itt is értelmezi az üzeneteket, mind a PC felé küldött, mind pedig a PC-től érkezőket, és eszerint továbbítja. Mint a robotnál, ugyan ez a probléma merült fel itt is. Tehát itt is egyszerűbb volt minden adatot egy stringbe összesítve továbbítani, mint már eleve CAN üzenetekre szétbontva.

5.5 Programfejlesztés

Programozás szempontjából a mikrokontrollerek programjainak felépítése megegyezik, kivéve az egyes technológiai folyamatokat, mivel mindegyik állomás más-más funkcióval rendelkezik. Lényeges egyezőségek között megemlítenénk a CAN kommunikáció inicializálását, valamint a beérkező CAN üzenetek megszakításban történő lekezelését, továbbá az egyéb, MPS állomás vezérléséhez szükséges függvényeket, amelyek egyszerűbbé teszik a programozást és az áttekinthetőséget.

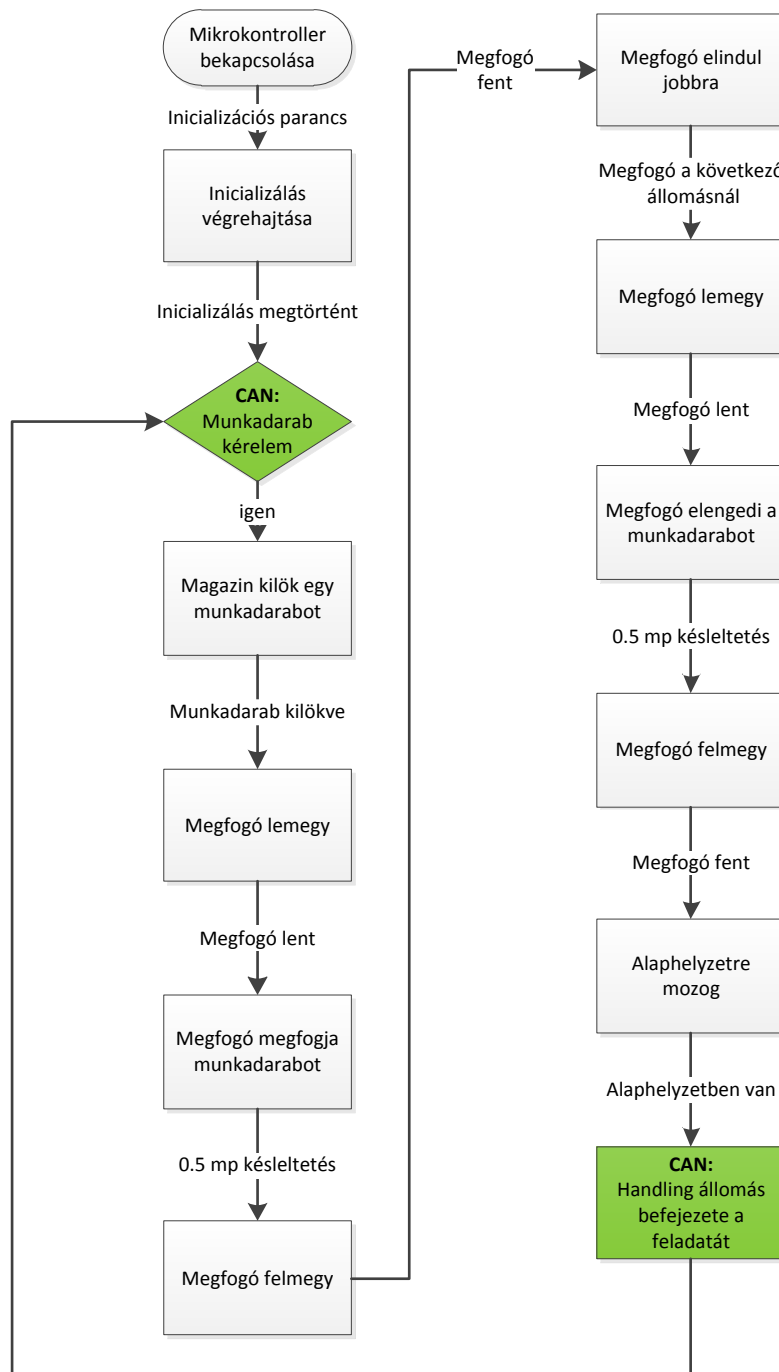
A továbbiakban a könnyebb megértés érdekében nem programrészleteket mutatunk be, hanem a folyamatábrákat, amelyek segítségével az adott állomások működő programjait elkészítettük.

5.5.1 Handling állomást vezérlő mikrokontroller programja

A Handling programja az alábbi folyamatábra alapján készült. Az állomást villamos feszültség és pneumatikus nyomás alá helyezve legelőször egy általános inicializáló folyamat megy végbe, ahol a végrehajtóit alaphelyzetbe állítjuk. Erre azért van szükség, mert nem biztos, hogy a megelőző munkafolyamat szakszerűen lett leállítva, és az állomás minden aktuátora alaphelyzetben van.

Handling állomás alaphelyzete:

- Kitoló munkahenger behúzva
- Megfogó nyitva
- Megfogót tartó munkahenger behúzva
- Lineáris hajtás kocsija a magazin fölött



21. ábra Handling állomás programjának folyamatábrája

Az alaphelyzet felvételét követően az állomás gyakorlatilag üzemkész állapotban van. Ekkor már csak CAN buszon keresztül várja az utasítást, hogy adagoljon egy munkadarabot. Az üzenet megérkezésével a magazin kitoló munkahengere kilök egy munkadarabot az ejtőtárból, majd a megfogó lemegy alsó helyzetébe és bezár. Egy kis időzítés után, a megfogó a munkadarabbal felmegy felső helyzetébe majd az elektromos lineáris hajtás átmozgatja a

következő állomás felett lévő pozícióba. Itt a megfogó újra lemegy, majd a Cap állomás szállítószalagja felett elereszti a munkadarabot. Rövid időzítést követően a megfogó felső helyzetébe mozog, majd felveszi a már említett alappozícióját. Az alappozíció sikeres felvétele után jelez a PC oldali mikrokontrollernek, hogy sikeresen adagolt egy munkadarabot.

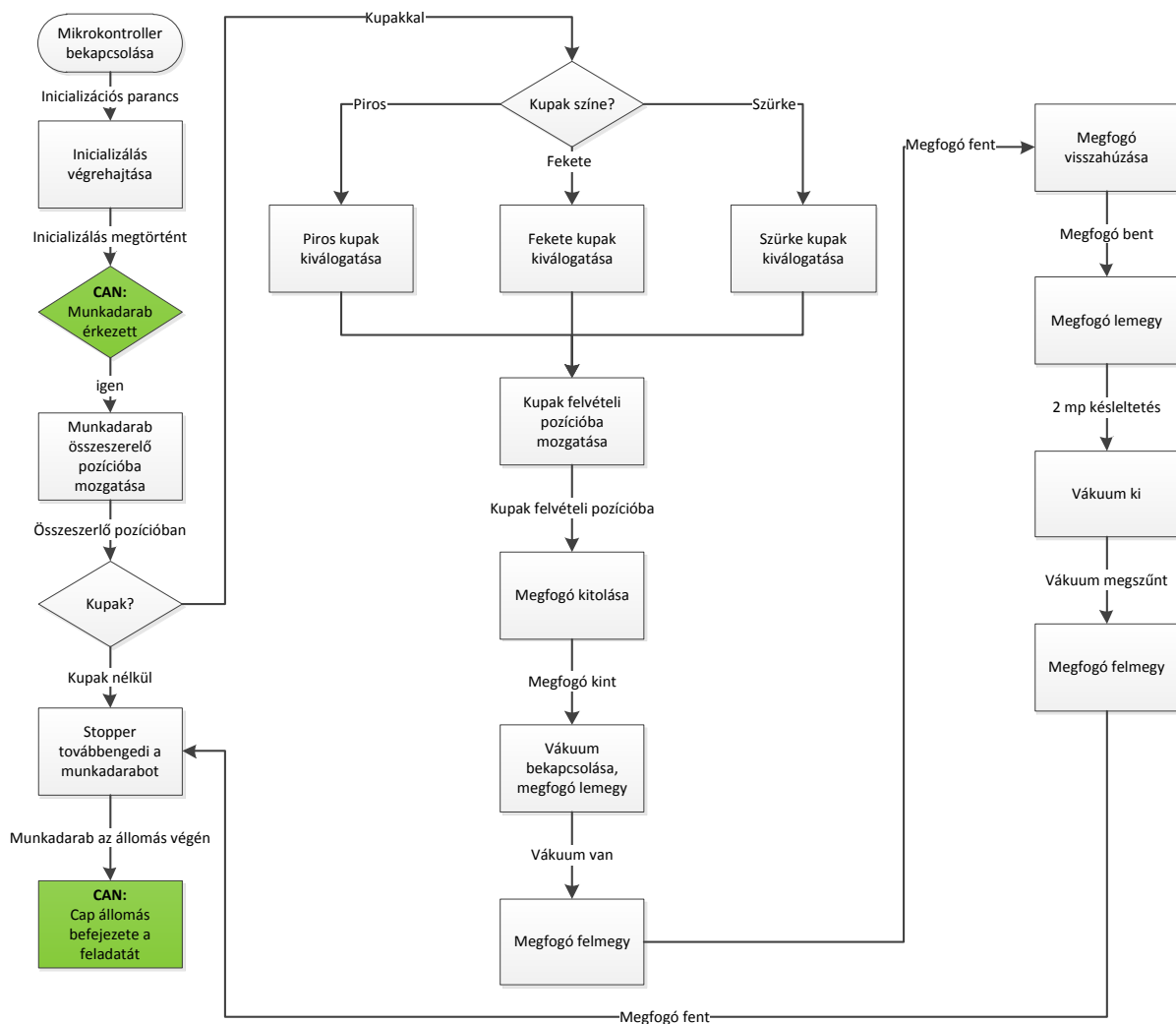
5.5.2 Cap állomást vezérlő mikrokontroller programja:

A Cap programja az alábbi folyamatára alapján készült. Az állomást villamos feszültség és pneumatikus nyomás alá helyezve legelőször egy általános inicializáló folyamat megy végbe, ahol végrehajtóit alaphelyzetbe állítjuk.

Cap állomás alaphelyzete:

- Stopper fordítómű munkadarabot fogad
- Szállítószalagok állnak
- Vákuum kikapcsolva
- Horizontális munkahenger behúzva
- Vertikális munkahenger behúzva

Alaphelyzet felvételét követően a berendezés üzemkész. Ekkor a CAN buszról várja az utasítást. A vétel után elindul a fő szállítószalag és a munkadarabot a Pick&Place modulhoz mozgatja. Amennyiben rendelés alapján szükség van kupakra, a vezérlő a két szállítószalag és színszenzor segítségével megkeresi a megfelelőt, majd felvételi pozícióba mozgatja. Ezután H1 munkahenger kimegy, a vákuum megfogó bekapcsolódik és a V1 munkahenger alsó helyzetébe mozog. Amint kialakult a vákuum és a vákuumdetektor jelez, megkezdődik az összeszerelés. A megfogót függőleges irányban mozgó V1 munkahenger felmegy, majd a H1-es belső helyzetébe mozog, így a kupakot a munkadarab fölé mozgatva. A V1 munkahenger újra lemegy, ezzel a kupakot ráhelyezi a munkadarabra, majd a vezérlő kikapcsolja a vákuumot és V1 visszatér alaphelyzetébe. A munkadarabot mozgó szállítószalag elindul és a stopper fordítómű továbbengedi az elkészült gyártmányt. Miután a termék lekerült a szalagról, az állomás felveszi alappozícióját és jelez a PC-oldali vezérlőnek, hogy sikeresen továbbította az elkészült munkadarabot a Robot számára.



22. ábra Cap állomás programjának folyamatábrája

5.5.3 Robot állomást vezérlő mikrokontroller programja

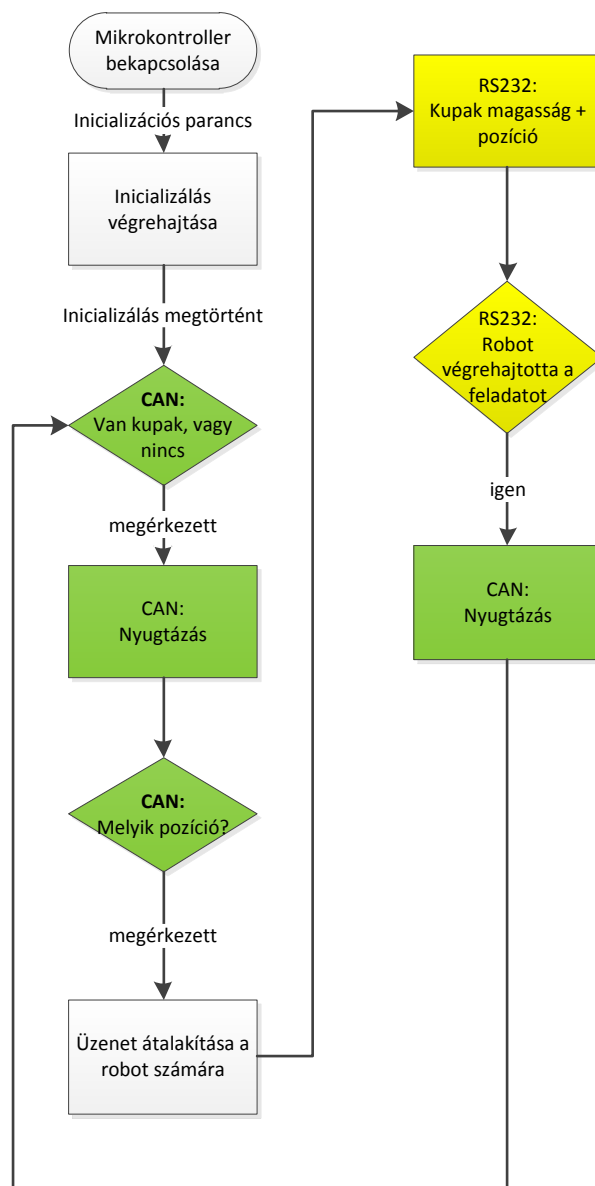
A robot mikrokontrollerének programja az alábbi folyamatábra alapján készült. Az állomást villamos feszültség és pneumatikus nyomás alá helyezve legelőször egy általános inicializáló folyamat megy végbe.

Robot állomás alaphelyzete:

- Robot a köztes P11 pozíciójában áll
- Robot megfogója nyitva

Az alaphelyzet felvételét követően a robot állomás mikrokontrollere várja a CAN buszról érkező utasítást. Robot megfogópofájának kialakítása miatt, a felvételi pozíció függ attól, hogy a gyártmányon van-e kupak, vagy sem, éppen ezért elsőként ez az információ érkezik. Nyugtázást követően a PC oldali mikrokontroller elküldi a paletta számát, ahova a robotnak le kell tennie a kérdéses munkadarabot. A robothoz tartozó vezérlő az üzenetet átalakítja úgy,

hogyan az a robot számára értelmezhető legyen, majd elküldi soros porton. Ezt követően a robot megfogja a fészekben lévő terméket, majd a palettára helyezi. Palettázás után a robot visszatér az alappozíciójába, majd soros porton visszajelez, hogy elvégezte feladatát. Ezután a robot mikrokontrollere jelez CAN buszon, hogy az állomás végzett.



23. ábra Robot állomás programjának folyamatábrája

5.5.4 PC oldali mikrokontroller programja

A PC oldali mikrokontroller programja az alábbi folyamatábra alapján készült el. Mikrokontroller bekapcsolását és inicializálást követően, a vezérlő várja a számítógépes

felügyeleti alkalmazástól a megrendelést. A megrendelés üzenete tartalmazza a megrendelendő gyártmányok darabszámát.

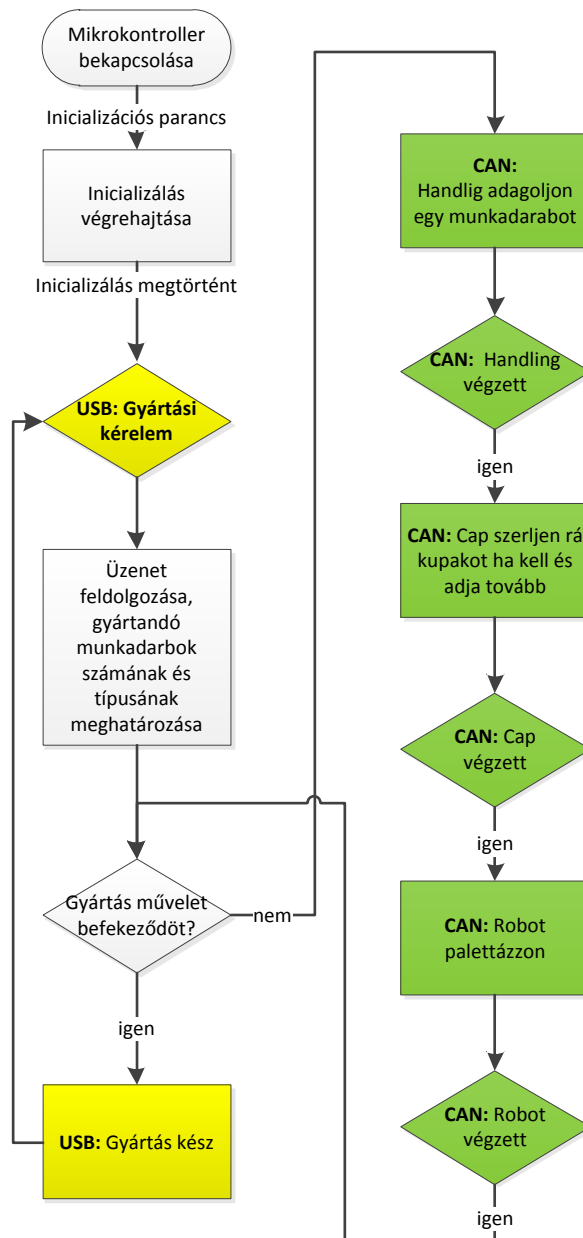
pl: „md1111” :

- 1. szám: 1db fekete munkadarab kupak nélkül
- 2. szám: 1db fekete munkadarab piros kupakkal
- 3. szám: 1db fekete munkadarab fekete kupakkal
- 4. szám: 1db fekete munkadarab szürke kupakkal

Az üzenet feldolgozást követően a vezérlő végrehajtja a következő ciklikus programrészletet:

- CAN buszon kér a Handling-től egy munkadarabot, majd várja a választ, hogy az teljesítette a feladatát.
- Ezt követően elküldi a Cap-nek, hogy a beérkező munkadarabra szereljen –e kupakot és ha igen akkor milyen színűt. Ezt követően várja választ, hogy az állomás teljesítette feladatát.
- Harmadik lépésben a robot számára elküldi, hogy az érkezett munkadarab rendelkezik-e kupakkal, valamint az a palettán hova kerüljön. Ezt követően várja választ, hogy az állomás teljesítette feladatát.

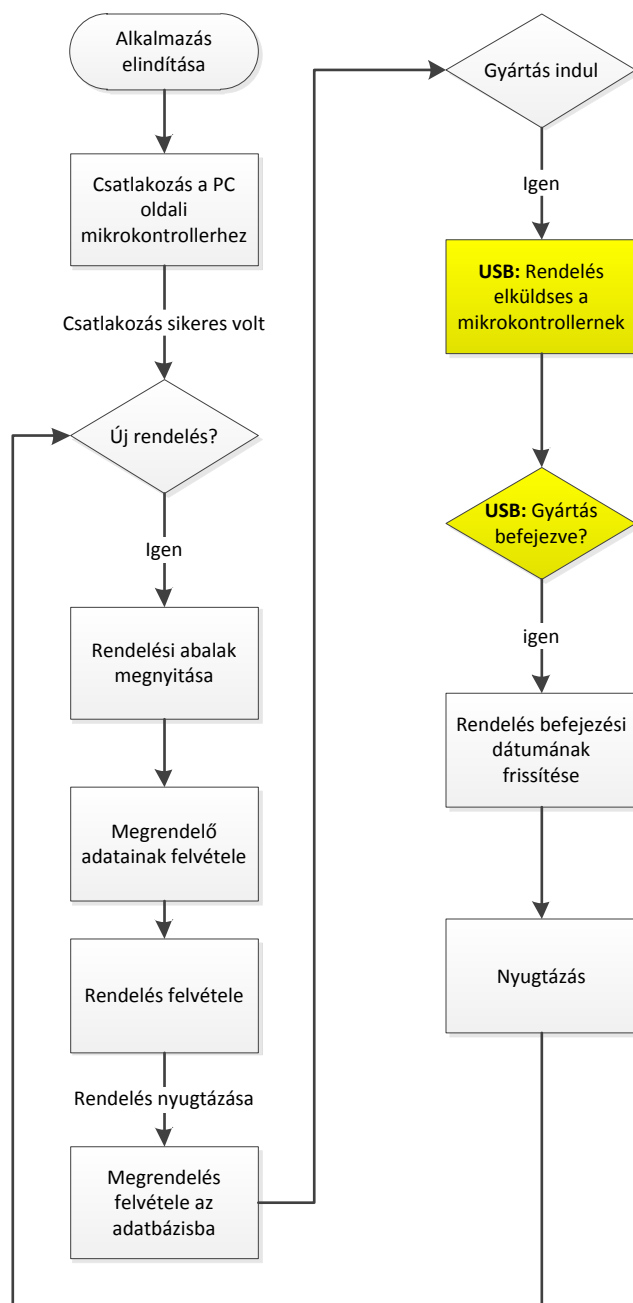
Amennyiben a megrendelésnek megfelelően, a legutolsó munkadarabot is legyártotta, értesíti a felügyeleti programot, hogy a gyártási folyamat kész.



24. ábra PC oldali mikrokontroller programjának folyamatábrája

5.5.5 PC felügyeleti program

Az általunk készített felügyeleti programot C# nyelven írtuk. A program lényeges részét képezi az USB-n keresztül történő kommunikáció a PC oldali mikrokontrollerrel, mivel ezen a csatornán keresztül tudjuk leadni a rendelésünket a gyártósor számára. Fontos még megemlíteni az adatbázis kezelést, amely segítségével nyomon követhető a megrendelő neve, rendelése, rendelésének ideje, rendelés legyártásának befejezési dátuma, stb. Terjedelmi okok, valamint könnyebb megértés érdekében a kódrészlet helyett, azt a folyamatábrát ismertetnénk, amely segítségével a felügyeleti program elkészült.



25. ábra Felügyeleti alkalmazás programjának folyamatábrája

6. Megoldás elemzése

Az elkészült rendszert összevetettük egy PLC-vel megvalósított vezérléssel. A megvizsgált szempontok eltérnek mind gazdasági, mind megbízhatósági, mind pedig a fejleszthetőségi problémákra az egyes eszközöknél.

Költség

A legnagyobb eltérés a vezérléshez felhasznált eszközök költségében mutatkozik meg. Egy beágyazott rendszerekből felépített vezérlés beszerzési ára töredéke a PLC-knek. Az általunk használt vezérlés alkatrészeinek költsége körülbelül 30-40eFt. Míg ugyan ez, az olcsóbb kategóriába tartozó kompakt PLC-kből felépítve, több 100e Ft lenne. Ez az árkülönbség jelentős előnyt jelent a beágyazott rendszerek javára, de következőkben látni fogjuk, hogy erre felszámolódhatnak még egyéb költségek, amelyeket a PLC-k ára már tartalmaz.

Fejlesztési idő

A másik fontos szempont a fejlesztési idő. Az ipari vezérlők esetén, egyszerűen összeállítható a konfiguráció, a legtöbb gyártó ebben támogatást is nyújt a vásárlóinak. A beágyazott rendszerek ezzel szemben célspecifikus berendezések, alkalmazásonként egyedi megvalósítás szükséges. Ezzel jelentősen növelve a fejlesztés idejét és költségeit (beleértve az esetleges prototípusok költségeit). A szoftver megvalósítása is lényegesen egyszerűbb egy PLC-re, a gyártó által biztosított program modulokkal, funkcióblokkal. Ezek elvégzik a különböző perifériák, kommunikációk kezelését, a fejlesztőnek nem kell részletesen ismernie a hardver felépítését. Ezek hiánya a beágyazott rendszereknél többszörösére növeli a fejlesztési időt. A cél specifikus eszközökből adódóan, egyes projektenként más-más megoldást alkalmaznak. A vezérléshez szükséges perifériákat nekünk kell lekezelni, saját kommunikációs protokollokat kell létrehozni, és részletesen ismerni kell a használt eszköz felépítését, működését. Ez különböző gyártók processzorainál akár lényeges eltéréseket is jelenthet.

Bővíthetőség

Beágyazott rendszerek segítségével szinte bármilyen periféria lekezelhető. Legyen szó egy egyszerű I/O kezelésről, vagy mérésadat gyűjtésről, képfeldolgozásról, motor fordulatszám szabályzásról, vezeték nélküli kommunikációról, RFID olvasóról vagy akár GPS-es

pozicionálásról. Fontos kiemelni, hogy a fenti technológiai modulokra is van a gyártóknak PLC-hez csatlakoztatható, iparban tesztelt megoldása. Kérdés a rugalmasság és az ára, mely egy vagy két nagyságrenddel is nagyobb lehet utóbbi esetben.

Számítási teljesítmény

A megfelelő processzort választva, olcsón megvalósíthatók a nagy számításigényű feladatok is. Ilyenek lehetnek például egy képfeldolgozó algoritmus, vagy egy összetettebb szabályozás. Ezen feladatok ellátására a legtöbb PLC nem képes, csak drága folyamatirányító számítógépeket alkalmazásával oldható meg.

Szabványosság

Az egyedi fejlesztésű vezérlők tervezése során figyelniük kell a hatályban lévő szabványok betartására. Ami szintén megnöveli a fejlesztés idejét, és a fejlesztéshez szükséges ismereteket. Ezen tanúsítványok megléte nélkül, legtöbb esetben nem is lehet forgalomba hozni az adott terméket. Ilyen előírások például a RoHS vagy a CE megfelelés. A termékek bevizsgáltatása, a megfelelések megszerzése külön költségeket jelenthet.

Minőségbiztosítás

Az ipari alkalmazások számára a PLC jelentős előnye, hogy a gyártó garantálja a berendezés élettartamát, biztonságos üzemelését, valamint a megfelelő mechanikai védelmet (IP védettségi szint) és az elektronika üzembiztos működését ipari környezetben is. Ezzel szemben a kisebb fejlesztő cégek csak komoly referencia munkák felmutatásával tudják erről meggyőzni a vevőiket, ha egyáltalán biztosítani tudják ezeket.

Támogatás

A PLC-k mellé a gyártók a garancián túl biztosítják a megfelelő támogatást is. Így egy esetleges meghibásodás, vagy valamely felmerülő probléma megoldásában a segítség biztosított.

Az előbbi szempontokat összegezve egyértelműen nem állapítható meg, hogy melyik vezérlőberendezés jobb folyamatautomatizálási feladatokra. Látható, hogy minkét megoldás kompromisszumokat hordoz önmagában, amelyeket még a tervezési fázisban mérlegelni kell, majd ennek alapján meghozni a megfelelő választást.

7. Továbbfejlesztési lehetőségek

A jövőben szeretnénk munkánkat továbbfejleszteni, amely többek között eleget tesz az Industry 4.0 néhány követelményének.

Szeretnénk a gyártósori berendezésünket további MPS állomásokkal bővíteni, melyek segítségével újabb technológia folyamatokat illeszthetünk rendszerünkbe. Ezek közül a legfontosabb egy automatizált magas raktár állomás, amely palettákat raktároz egy 3x4-es polcrendszeren. Továbbá a tálcák RFID tag-ekkel lennének ellátva, így egyes leolvasásokat követően a számítógépes felügyeleti programunk kijelezné a hozzátartozó megrendelő személy adatait, valamint a palettán lévő gyártmányok különféle paramétereit és gyártási adatait.

Tervben van egy operátor panel fejlesztése, melynek segítségével a kezelőszemély tájékozódhat a legyártandó-, legyártott-, esetleg a selejtes munkadarabok számáról. Továbbá a gyártósori berendezés esetleges meghibásodása esetén instrukciókkal segítené a karbantartó személyzetet a hiba helyéről, mértékéről, hogy minél előbb folytatódhasson a gyártási folyamat.

Szeretnénk még a munkadarabokat is RFID tag-ekkel ellátni, hogy a gyártási folyamat során az MPS-ek az RFID chippek tartalmát kiolvassa, meghatározzák a szükséges gyártási lépéseket, majd megtervezzék azok legoptimálisabb végrehajtási sorrendjét. Ezzel egy önoptimalizáló rendszert létrehozva.

Továbbá szeretnénk, hogy egy webserver segítségével lehetővé tegyünk, hogy rendelési folyamatokat tetszőleges eszközökről, bárhol lebonyolíthassuk. Ennek procedúrája egy előzetes regisztráció után mehetne végbe, ahol a megrendelő megadná nevét, lakcímét, szállítási adatit, stb. Ezek után a rendelést megadva, valamint a lényeges fizetési eljárás után kezdődne meg a gyártás.

8. Köszönetnyilvánítás

Köszönjük Sándor Tamás Tanár Úrnak, amiért segített bennünket szakmai kérdésekben, valamint figyelemmel kísérte munkánkat.

Köszönet a Festo Kft.-nek, amiért rendelkezésünkre bocsátotta a felhasznált eszközöket, valamint a Festo munkatársainak Szabó Norbertnek és Bolla Dánielnek a szakmai hozzájárulást.

Továbbá köszönjük a szakmai segítséget Román Szabolcs és Eibeck Tamás egyetemi hallgatóknak.

9. Irodalomjegyzék

- [1] Kovács Csongor – Elektronikus áramkörök, General Press Kiadó, 1997
- [2] Tukora Balázs: Robotok Irányítása Pécs, 2004
- [3] Bolla Dániel - Robot Basics, FESTO jegyzet, 2013
- [4] ELEKTRONET - XXII. évfolyam 5. szám
- [5] http://www.festo.com/cat/hu_hu/products
- [6] <http://www.festo-didactic.com/hu-hu/oktatasi-eszkoezoek/mps-the-modular-production-system/mps-200/?fbid=aHUuaHUuNTUzLjM0LjIwLjg1Ng>
- [7] <http://www.intersil.com/content/dam/Intersil/documents/icl2/icl232.pdf>
- [8] http://www.st.com/web/en/resource/technical/document/application_note/CD00165404.pdf
- [9] <http://www.atmel.com/Images/doc7679.pdf>
- [10] <http://www.st.com/web/en/resource/technical/document/datasheet/CD00059620.pdf>
- [11] <http://msdn.microsoft.com/en-us/default.aspx>
- [12] <http://www.isocom.com/optocouplers>
- [13] <http://www.rixan.com/Portals/0/RV-1A-2A/1n2specs.pdf>