

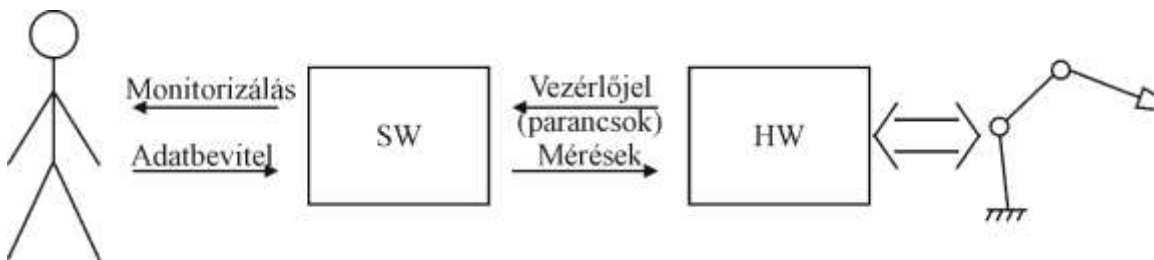
SCARA robot programozása

1. A gyakorlat célja

Az MTAB SCARA robot programozási nyelvének megismerése. Robotprogram fejlesztése cilindrikus munkadarab áthelyezésének megoldására.

2. Elméleti bevezető

A robotok felhasználói általában robotprogramozási nyelv segítségével írják le a robot által megoldandó munkát. A felhasználónak nem szükséges ismerni, hogy a robotirányítási rendszer milyen pályatervezési illetve irányítási algoritmust alkalmaz, a robotprogramozási nyelv egy egyszerű megoldást biztosít a robot által elvégzendő feladat megoldására.



1. Ábra: Felhasználó – robot kapcsolat

A feladat megoldása általában két lépésben történik:

1. A felhasználónak lehetősége van kézi vezérlésre – a robotot a munkatér különböző pontjaiba mozgatja – *betanítás* – a munkafázisoknak megfelelő pontokat elmenti a memóriába
2. Megírja a programot, amely a robotot a munkafázisoknak megfelelően végigvezeti a pontokon

2.1 Az ARPS robotprogramozási nyelv

Egyik közismert programozási nyelv az un. ARPS programozási nyelv, amelyet PUMA típusú robotokra dolgoztak ki. A következőkben bemutatjuk a fontosabb, robotokkal kapcsolatos argumentumait, utasításait.

A program által alkalmazott argumentumok:

Egész állandó és változó

nnn -32768 .. 32767

var -32768 .. 32767

szimbolikus név

Decimális állandó

	nn.n		-3276.8 .. 3276.7
	n.nn		-327.68 .. 327.67
<i>Távolság</i>	dis		-1024.00 .. 1023.99
<i>Szög</i>	ng		-180.000 .. 179.995
<i>Csukló index</i>	jnt		1 .. 6
<i>Sebesség</i>	n.n		2 .. 500 mm/s
<i>Pontok</i>	loc		szimbolikus név
	precíz pont	#PONT	csuklókoordinátákban értendő
	koordinátpont	PONT	P=x,y,z,a,o,t - alap koordináta rendszerben
	kombinált pont	BASE(PONT)	egy bázispontnak és az ehhez képest relatív pontnak
			az eredő tranformációja
	inverz pont		a pontnak, mint transzformációnak az inverze
			(pontból az alap koordináta rendszer origójába)
	tükrőpont		relatív pont egy bázisponthoz képesti speciális

Pontokkal kapcsolatos utasítások

HE	HERE loc	az aktuális robothelyzetet a megadott néven eltárolja a memóriában, 'loc' lehet precíz, koordináta vagy relatív pont
LOC	LOCATE loc=[INVERSE] loc	precíz, koordináta vagy relatív pont pont másolása, átalakítása, 'loc' lehet
F	FRAME loc=loc,loc,loc[,loc]	üzemi keret megadása, paraméterek sorrendben: origó, x tengely pontja, xy sík pontja, eltolt (végleges) origó
SH	SHIFT loc=[dis],[dis],[dis]	pont eltolása x, y, z tengelyek mentén
DIST	DISTANCE var=loc,loc	két pont távolságát adja meg, az eredmény egész!
MI	MIRROR loc	tükröző pontot definiál
N M	NO MIRROR	a tükrözést tiltja
SC	SCALE loc=var,var,var	skálázás bázispontját definiálja, például 'SCALE BASE=2000,2000,2000' után 'BASE(A)' esetén A pont mindhárom tengely irányában kétszer olyan messze lesz BASE-től, mint skálázás nélkül
N S	NO SCALE	a skálázást tiltja

Program elágaztatások

JU	JUMP lbl	feltétel nélküli ugrás a megadott címkére
IF [INGROUP] var cmp [INGROUP] var THEN JUMP lbl		feltételes ugrás
IF IN var[,var][,var][,var]		digitális bemenetek állapotától függő feltételes ugrás (maximum 4 bemenet ÉS kapcsolata)
CA	CALL prg	szubrutinhívás (paraméterátadás nincs, minden változó és pont globális)
RE	RETURN [var]	szubrutinból való visszatérés, ha van 'var', akkor annyi számú programsort kihagy a hívó programból visszatéréskor

Mozgató utasítások

GO	GO loc	a megadott pontba megy interpolált pályán, speciális eset: GO READY - 'home' pozícióba megy
GOS	GOS loc	egyenes pályán megy a kívánt pontba
GO&O	GO&O loc	nyitja a megfogót, majd a megadott pontba megy interpolált pályán
GO&C	GO&C loc	zárja a megfogót, majd a megadott pontba megy interpolált pályán
GOS&O	GOS&O loc	nyitja a megfogót, majd a megadott pontba megy egyenes pályán
GOS&C	GOS&C loc	zárja a megfogót, majd a megadott pontba megy egyenes pályán
GON	GONEAR [loc],dis	abba a pontba megy interpolált pályán, amelytől a megadott pont a szerszám z tengely irányában az adott távolságra van, ha a pontot nem adjuk meg, az aktuális pontból megy a szerszám -z tengelye irányában az adott távolságra
GOSN	GOSNEAR [loc],dis	abba a pontba megy egyenes pályán, amelytől a megadott pont a szerszám z tengely irányában az adott távolságra van, ha a pontot nem adjuk meg, az aktuális pontból megy a szerszám -z tengelye irányában az adott távolságra
GOS&W	GOS&WEAVE loc	a megadott pontba megy egyenes pályán és közben lengő mozgást végez a szerszám xy síkjában
MOVE	MOVE [dis],[dis],[dis]	relatív mozgás az alap koordináta rendszer x, y, z irányában interpolált pályán
MOVES	MOVES [dis],[dis],[dis]	relatív mozgás az alap koordináta rendszer x, y, z irányában egyenes pályán
TMOVE	TMOVE [dis],[dis],[dis]	relatív mozgás a szerszám koordináta rendszer x, y, z irányában interpolált pályán
TMOVES	TMOVES [dis],[dis],[dis]	relatív mozgás a szerszám koordináta rendszer x, y, z irányában egyenes pályán
M J	MOVE JOINT jnt,ang	a megadott csuklót az adott szöggel mozgatja
A	ALIGN	a szerszám z tengelyét a legközelebbi alap koordináta tengely irányába állítja úgy, hogy közben a pozíció változatlan marad (tengely irányú csavarodás marad)

Sebesség beállítása

SP	SPEED nn.n	aktuális sebesség megadása [mm/s]
SP%	SPEED% var	sebességtényező megadása (3..300)
SP N	SPEED NEXT nn.n	a következő mozgató utasítást ezzel a sebességgel hajtja végre

Szerszám megfogó vezérlése

OP	OPEN	nyitja a megfogót
CL	CLOSE	zárja a megfogót
OD	ODELAY n.nn	az OPEN utasítást paraméterezi, nyitás után a megadott ideig vár
CD	CDELAY n.nn	a CLOSE utasítást paraméterezi, zárás után a megadott ideig vár

rendszerének xy síkjára, orientációja pedig az alap

Koordinátarendszer módosítása

B BASE [dis],[dis],[dis],[ang] alap koordinátarendszer transzformálása, a paraméterek sorrendben: dx, dy, dz, do
TOO TOOL [dis],[dis],[dis],[ang],[ang],[ang] szerszám koordinátarendszer transzformálása, a paraméterek sorrendben: dx, dy, dz, do, da, dt
LT LTOOL loc ugyanaz, mint az előbb, csak egy ponttal lehet megadnia transzformációt

Pontok meghatározása

CH CHANGE loc listázza a pont értékét, meg lehet változtatni (új pont generálása), loc: precíz- vagy koordináta pont
H HERE loc pont értékének bevitele a memóriába (aktuális robothelyzet szerinti új pont generálása), loc: precíz-, koordináta- vagy relatív pont
WH WHERE [#] folyamatosan listázza az aktuális pontot
LT LTEACH loc pontok betanítási üzemmód, a betanító pulton lévő STEP gomb megnyomásakor mindig egy új pont tárolódik el, loc: precíz-, koordináta- vagy relatív pont

Operátorok

Az összes „klasszikus” operátor értelmezett: +, -, *, /, <, >, =, AND, OR

2.2 Robotirányítási hardver és szoftver architektúrák

A robotirányításra kifejlesztett vezérlők kell biztosítsák a robot – felhasználó közötti kapcsolatot illetve, hogy a robot végrehajtsa az előírt feladatot. A hardver kell, hogy tudja fogadni a pozíció és más típusú érzékelőktől kapott jeleket, ugyanakkor az irányítóprocesszor nagy számításkapacitással kell rendelkezzen. A szoftvernél célszerű többszörös, többprocesszes megoldást alkalmazni, ugyanis vannak feladatok, amiket nagy prioritással kell elvégezni (például alacsony szintű irányítás) és vannak, amiket kis prioritással (például monitorizálás). A szoftvernek ugyanakkor tartalmaznia kell egy matematikai, mátrixműveleteket elvégző könyvtárat is, a robotgeometriai, kinematikai feladatok megoldására.

A hardver kialakításánál két megoldás terjedt el:

1. Központi vezérlővel (2. Ábra)

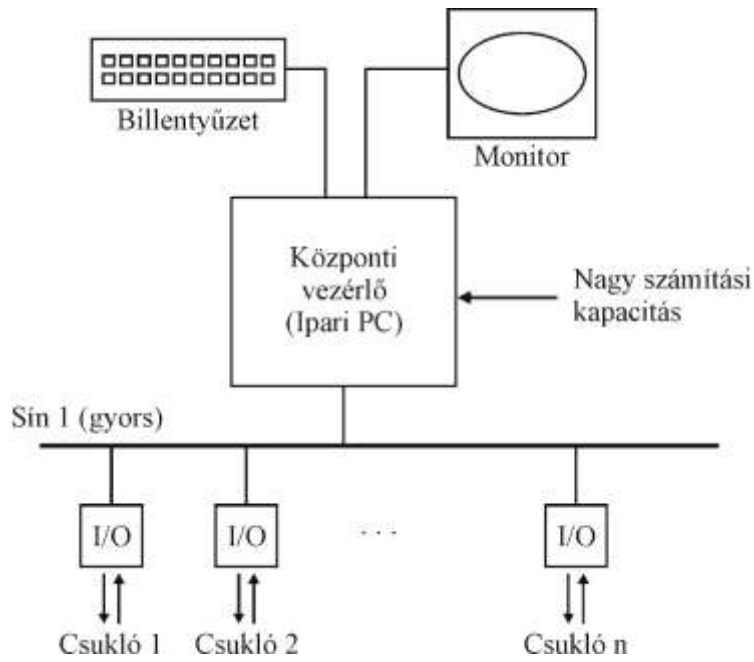
A teljes szoftver a központi vezérlőn fut: Robotprogramozás, Robotikai számolások (Kinematika, Geometria), Pályatervezés, irányítási algoritmus

Az I/O modulok csak a mérésekért felelősek, valamint kapcsolatért az erősáramú résszel

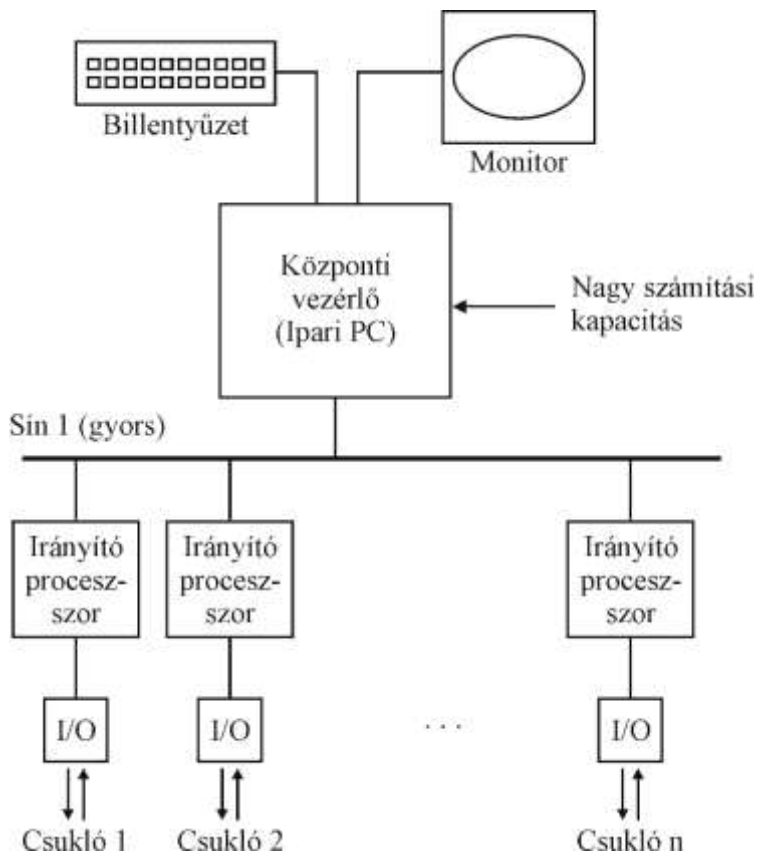
2. Elosztott felépítés (3. Ábra)

Központi vezérlőn fut a robotprogramozás, robotikai számolások (Kinematika, Geometria), Pályatervezés. Kiküldi az előírt értéket az irányító processzornak
Irányító processzoron fut az irányítási algoritmus

Mivel az irányító processzor csak a saját csuklójáról kap mérési eredményeket, az irányítási processzoroknál lehetőség kell legyen az egymással történő kommunikációra, ugyanis a csukló beavatkozó jele függhet a többi csukló pozíciójától sebességétől is.



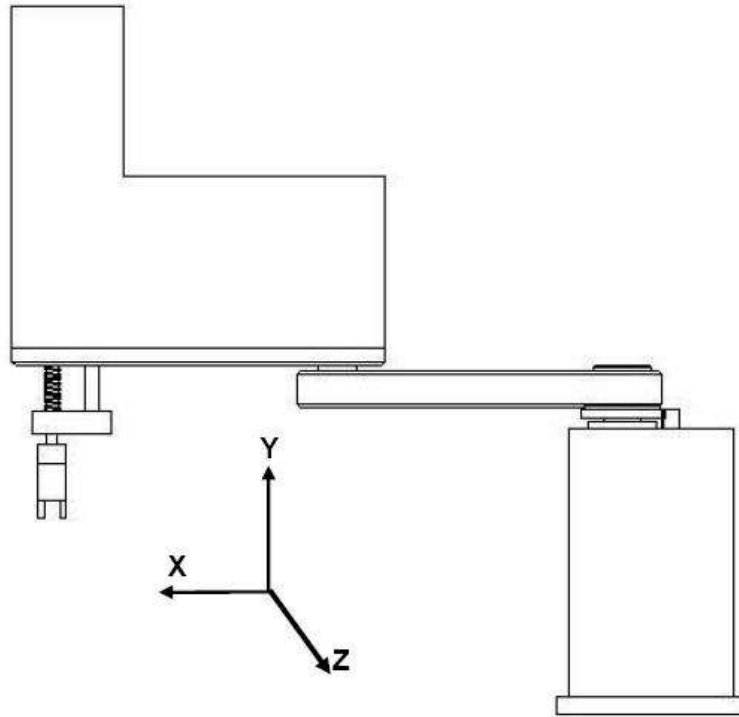
2. Ábra: Központosított robot hardver



3. Ábra: Elosztott robothardver

3. A mérés menete

Feladat: Legyen a 4 Ábrán látható MTAB SCARA robot. Íjunk egy programot, felhasználva a robot programozási nyelvét, amely segítségével a robot végberendezése egy, az asztallapon elhelyezett tárgyat az asztallapon egy másik előírt pozícióba elhelyez.



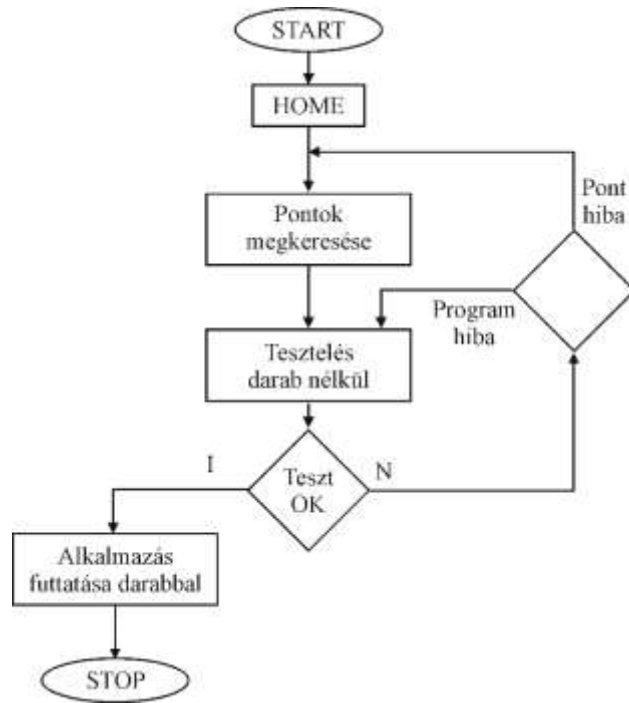
4. Ábra: MTAB SCARA kar

A feladat megoldásának menetét az 5 Ábrán látható folyamatábra mutatja. A feladathoz a robot munkaterében négy célpontnak a koordinátáit kell lejegyezzük:

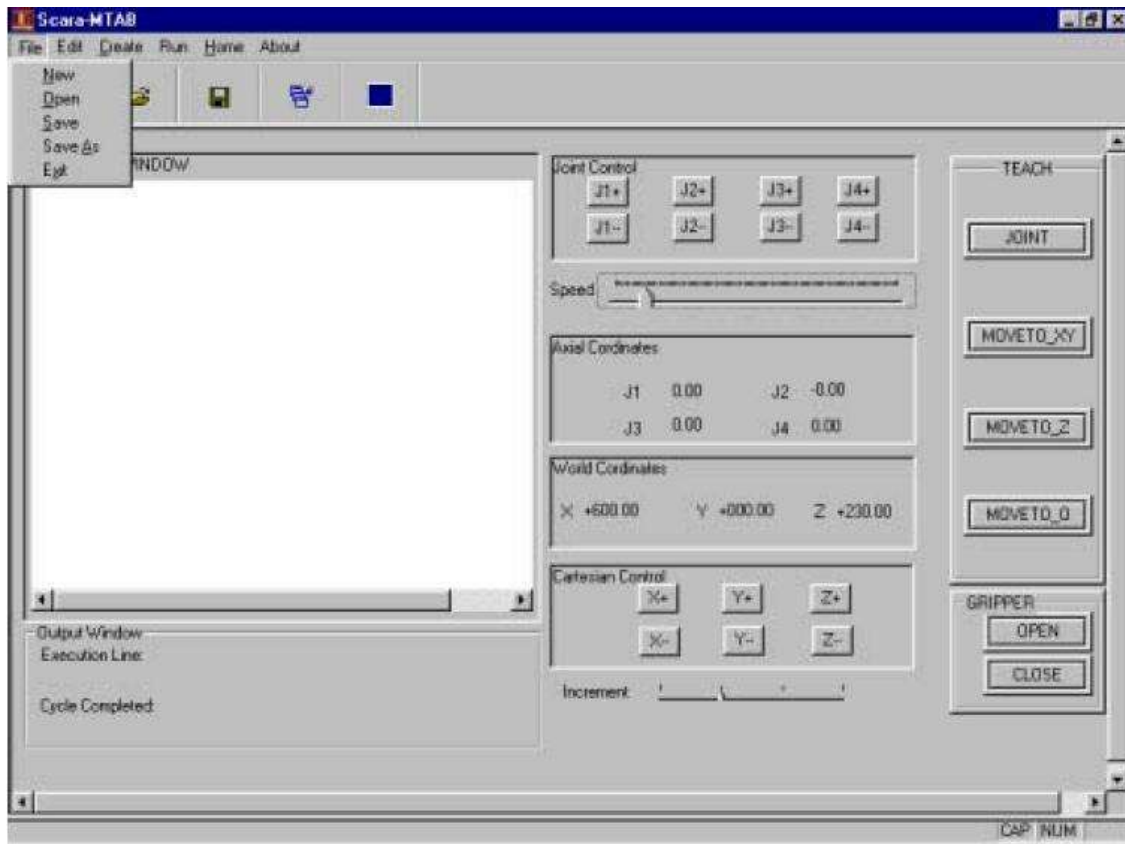
- kezdőpont
- kezdőpont felett
- célpont
- végpont felett

A pontok megkereséséhez a robotot kézi üzemmódban vezéreljük. Ehhez a robot felhasználói interfészén levő gombokat, illetve pozíció kiírást alkalmazzuk.

- Nullázzuk a robot pozícióját a HOME ALL utasítással.
- A robot kezébe adjuk a munkadarabot a GRIPPER OPEN, GRIPPER CLOSE gombok segítségével
- A sebességet (SPEED) 60-80 százalékosra állítjuk
- Eljuttatjuk a célpozícióba a végberendezést. Ehhez a JOINT CONTROL gombokat alkalmazzuk.
- A célpontban leolvassuk a csuklósögeket az AXIAL COORDINATES ablakból.
- Lejegyezzük a célpont koordinátáit.

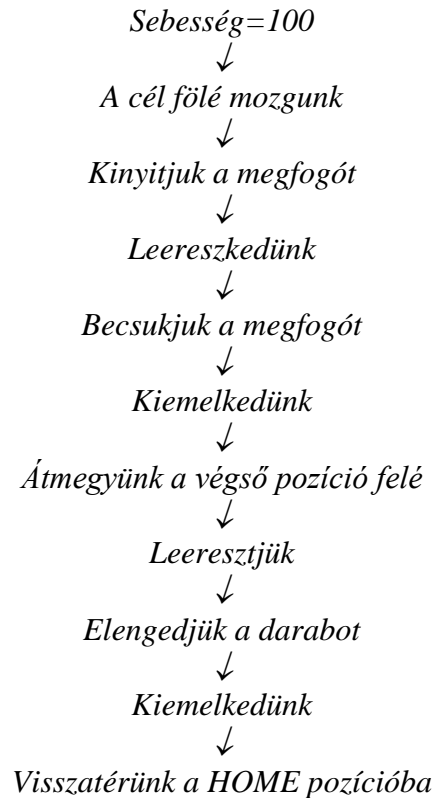


5. Ábra: A laboratóriumi gyakorlat folyamatábrája



7. Ábra: A program felhasználói interfésze

A program az alábbi szekvenciákat kell tartalmazza:



A program megírásához az alábbi utasításokat kell alkalmazni:

SPEED *value* – sebesség értékének megadása (*value* a 0 ... 100 tartományban van)

GRIPPER OPEN/ CRIPPER CLOSE – megfogó kinyitása, becsukása

JOINT A1 *val1* A2 *val2* A3 *val3* A4 *val4* – Egy adott pontba eljuttatja a robotot, amelyet a *val1* ... *val4* értékek definiálnak.

4. Kérdések és feladatok

1. Keressen az Interneten más robotprogramozási nyelveket. Hasonlítsa össze őket az MTAB SCARA illetve az ARPS robotprogramozási nyelvekkel.
2. Módosítsa a programot úgy, hogy a leírt feladatot háromszor hajtsa végre egymás után.
3. Oldja meg a feladatot csuklóváltozók helyett világkoordinátákban megadott változókkal, programutasításokkal.