

Programarea si simularea offline a unei celule flexibile robotizate pentru debavurare repere din industria auto utilizand mediul de lucru Kuka Sim. Programarea pe baza de semnale.

PROGRAMMING AND OFFLINE SIMULATION OF A DEBURRING CELL USING WORKING ENVIRONMENT KUKA SIM.SIGNAL-BASED PROGRAMMING.

Vasilescu-Dinca Loredana

Facultatea:I.I.R, Specializarea:Robotica, Anul:II, e-mail:vasilescu_loredana95@yahoo.com

Conducător științific: Prof.dr.ing. Nicolescu Adrian Florin

Summary: In this study was performed the offline programming of the robotic application for deburring. It is presented the logical organization of events and the definition of signals. Also its presented the offline simulation of functioning based on the signals application. In the robotic cell is deburred a secondary shaft from a gearbox.

1.Introducere

Obiectivele urmarite in aceasta lucrare sunt realizarea programarii offline a aplicatiei robotizate pentru debavurare. In celula robotizata se realizeaza debavurarea unui arbore secundar al unei cutii de viteze. Este prezentata organizarea logica a evenimentelor si definirea semnalelor. Aceasta se realizeaza in programul Work Visual 3.1. Este prezentata de asemenea simularea offline a functionarii pe baza de semnale a aplicatiei. Simularea este realizata in programul Kuka Sim 1.1.

2.Studiu actual

În figurile de mai jos. se prezintă structura de ansamblu a aplicației robotizate de proiectat respectiv vederea de sus de tip drafting pentru aplicația de proiectat, cu repere și legenda ce include denumiri pentru toate subsistemele.

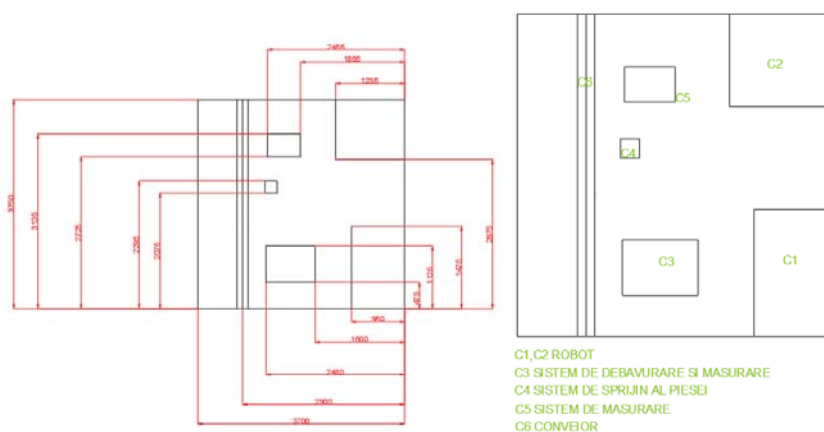


Fig.1.Vedere de sus de tip drafting pentru aplicația de proiectat

Descrierea funcționării de ansamblu a aplicației și a rolului funcțional specific al subsistemelor componente în cadrul acesteia. În aplicația studiată sunt incluși doi roboți cu arhitectură generală de tip braț articulată care manipulează piesele ce intră pe bandă în celulă. În interiorul celulei se află un dispozitiv de debavurare a canalelor piesei și un dispozitiv de înlăturare cu jet de aer a bavurii rămase. De asemenea se regăsește conveiorul pe care intră și ies piesele și un sistem perirobotic de susținere a piesei.Pentru realizarea temei au fost consultate referintele bibliografice :[1],[2],[3].

In urmatoarea imagine este prezentata celula robotizata studiata realizata in programul Kuka Sim. Aceasta celula contine un subsistem de debavurare,un subsistem de inlaturare a bavurii,doi roboti Kuka KR 60-3, un conveior,un subsistem de sprijin pentru piesa si controlerele aferente statiei.

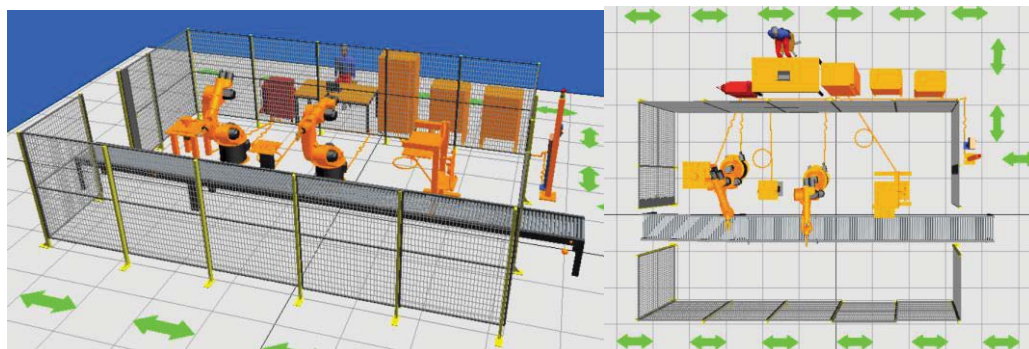


Fig.2.Prezentarea celulei robotizate vazute din lateral si vazuta de sus

Organizarea fluxurilor logice de evenimente si definirea semnalelor

Pentru definirea semnalelor este folosit mediul de lucru Kuka WorkVisual in care se configureaza in prima etapa controlerele. Cum aplicatia studiata foloseste doi roboti Kuka KR 60-3 se vor alege doua controlere KRC4.In continuare este prezentata crearea configurarea aplicatei si alegerea tipului de robot pentru aceasta si arhitectura aplicatiei vizualizata in mediul de lucru WorkVisual.

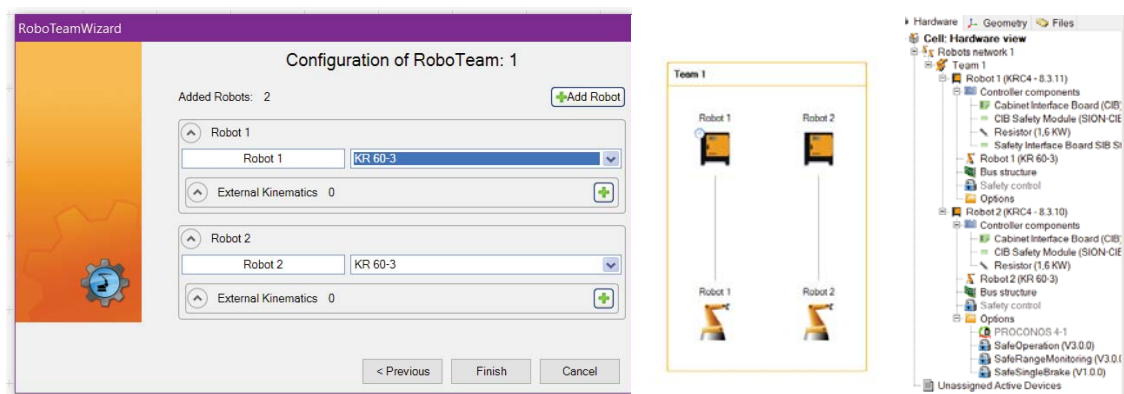


Fig.3.Configurarea arhitecturii celulei robotizate si prezentarea arhitecturii celulei robotizate

Cea mai importanta etapa este definirea semnalelor de intrare si iesire (digital input/digital output).In imaginea de mai jos sunt prezentate semnalele de iesire cu denumirea lor si prescurtarea.

Definirea semnalor este impartita in doua mari categorii si anume semnale digitale si analogice. La randul lor se impart in alte doua cateogrii si anume de intrare si de iesire.

Pentru definirea semnalelor avem nevoie mai intai de definirea succesiunii operatiilor din celula robotizata analizata.

In proiectul studiat avem urmatoarele operatii: oprirea paletei in dreptul R1; R1 ia piesa de pe paleta; plecarea paletei de la R1 la R2; masurarea si debavurarea piesei pe dispozitivul special; detectarea piesei cu ajutorul senzorilor pe suportul intermediar; R2 ia piesa de pe suportul intermediar; inlaturarea bavurii cu jet de aer; R2 pune piesa pe paleta; paleta pleaca

Aceste operatii se vor impartii in trei categorii de semnale si anume semnale de intrare(digital input),semnale de iesire(digital output) si semnale interne ale echipamentelor(station signal/flag). In continuare sunt prezentate aceste semnale ce se vor numerota cu litere alfabetice pentru a putea fi citite mai usor. Deoarece in proiectul studiat se regasesc doi roboti,se vor defini semnale diferite pentru fiecare din acestia.

Pentru primul robot au fost definite urmatoarele semnale prezentate in capturi de ecran din program:

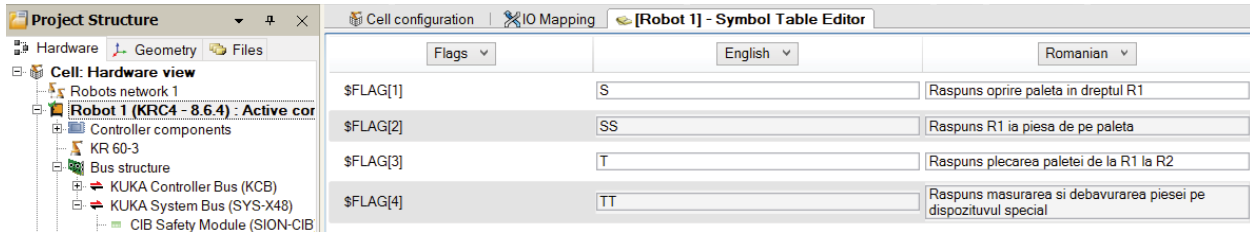


Fig.4. Semnale interne ale echipamentelor(station signal/flag)

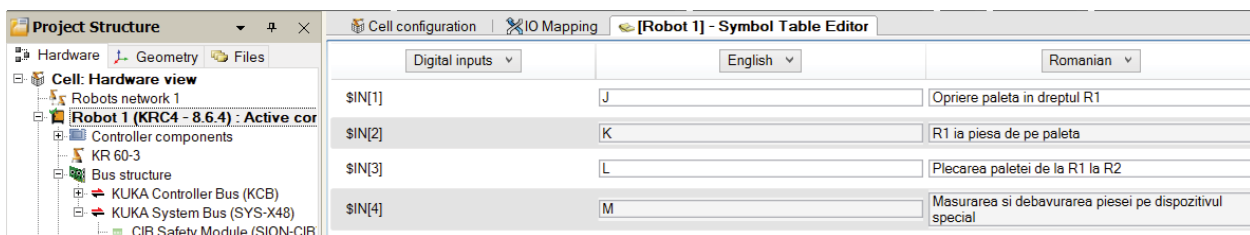


Fig.5. Semnale de intrare

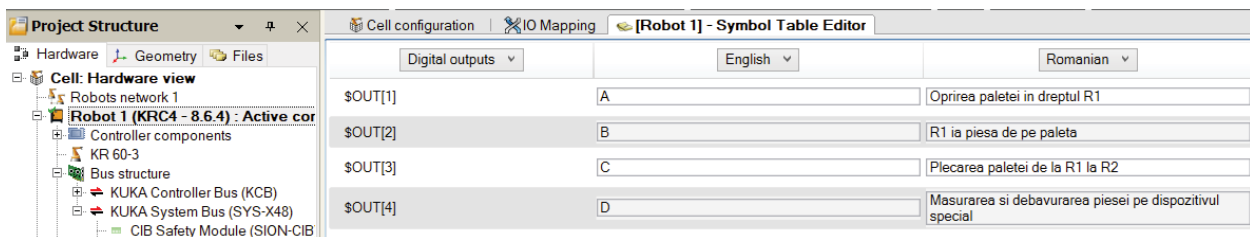


Fig.6..Semnale de iesire

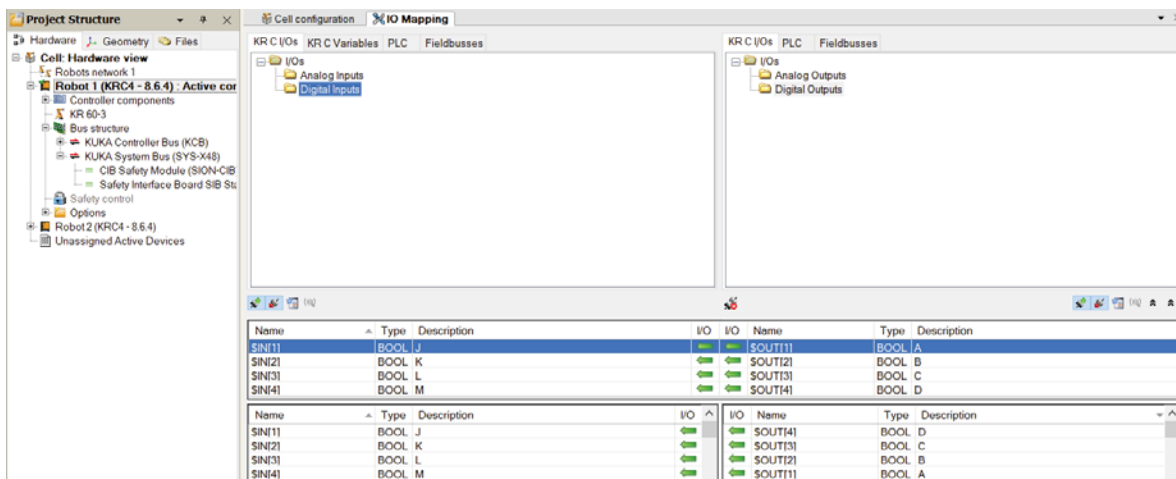


Fig.7.Prezentarea tabelii cu semnale DI/DO pentru primul robot

Pentru cel de-al doilea robot au fost definite urmatoarele semnale prezentate in capturi de ecran din program:

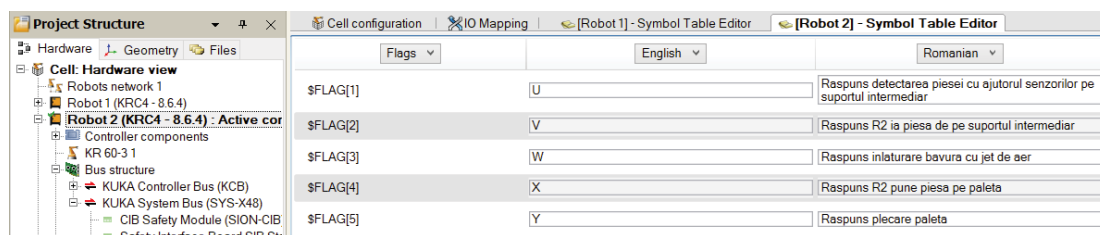


Fig. 8.Semnale interne ale echipamentelor(station signal/flag)

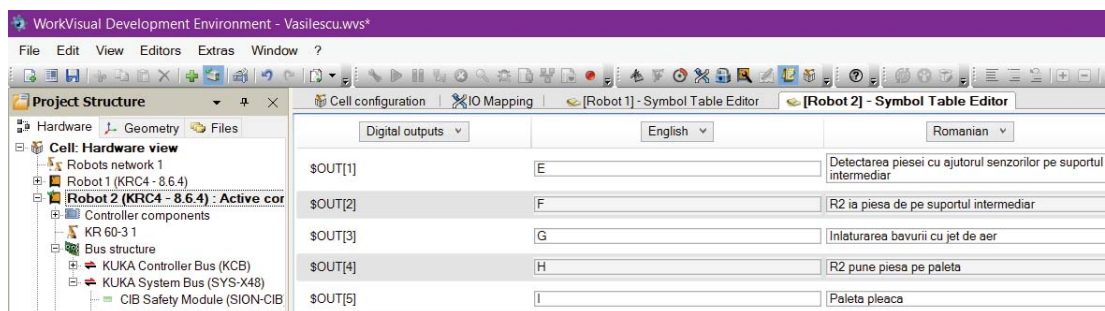


Fig.9.Semnale de iesire

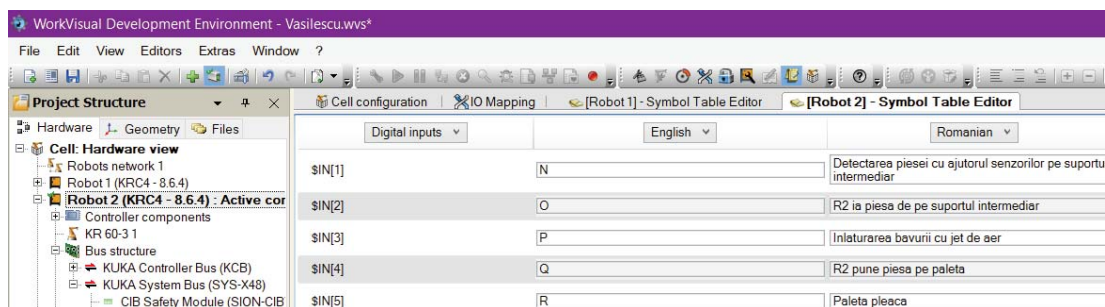


Fig.10.Semnale de intrare

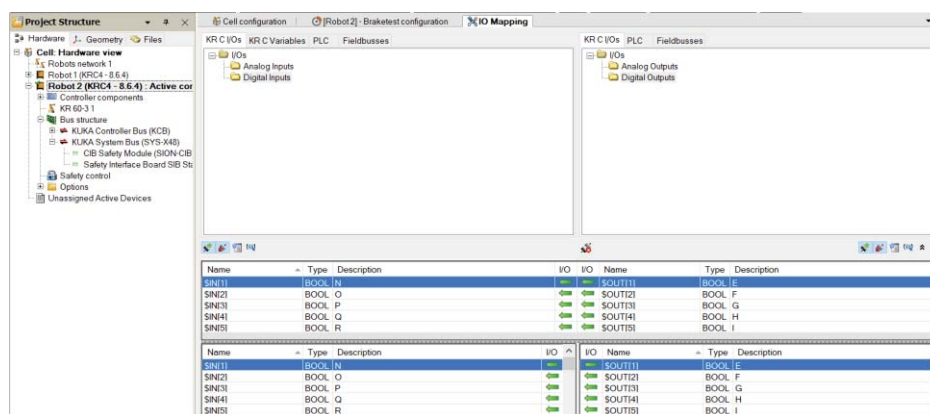


Fig.11.Prezentarea tabelii cu semnale DI/DO pentru cel de-al doilea robot

Se realizeaza in continuare definirea datelor despre robot si dispozitivele din celula cu care lucreaza . Aceasta definire este realizata pentru ambii roboti ea fiind aproximativ similara.Diferenta este ca la primul robot va lucra cu o masina in 4 axe iar cel de-al doilea cu o masina in doua axe.In aceasta se specifica numarul axelor,limitarile pe cuple etc.

```

1 SUCCESS(NT)
2 DEFDAT masina
3 ;FOLD EXTERNAL DECLARATIONS:$(PE)%MKUKATPBASIS,%CEXT,%VCOMMON,%P
4 ;FOLD BASISTECH EXT:$(PE)%MKUKATPBASIS,%CEXT,%VEXT,%P
5 EXT BAS (BAS_COMMAND :IN,REAL :IN )
6 ;ENDFOLD (BASISTECH EXT)
7 ;FOLD USER EXT:$(E)%MKUKATPUSER,%CEXT,%VEXT,%P
8 ;Make your modifications here
9
10 ;ENDFOLD (USER EXT)
11 ;ENDFOLD (EXTERNAL DECLARATIONS)
12 ENDDAT
13 &ACCESS RV
14 &PARAM DISKPATH = MADA
15 &PARAM VERSION = 1.0.0
16 DEFDAT masina PUBLIC
17 CHAR $V_RIMADA[32]
18 $V_RIMADA[]="V1.0.0/KUKA8.2" ;identificare versiune
19 INT STECH_MAX=4 ;numarul maxim de functii ale masinii
20 INT SNUM_AX=6 ;axele robotului
21 INT S_AXIS_TYPE[12] ;identificare axe
22 S_AXIS_TYPE[1]-3 :1 = liniare, 3 = rotative
23 S_AXIS_TYPE[2]-3
24 S_AXIS_TYPE[3]-3
25 S_AXIS_TYPE[4]-3
26 S_AXIS_TYPE[5]-3
27 S_AXIS_TYPE[6]-3
28 S_AXIS_TYPE[7]-1
29 S_AXIS_TYPE[8]-3
30 S_AXIS_TYPE[9]-3
31 S_AXIS_TYPE[10]-3
32 INT SVITEZA_VAL=50 ;Limitare viteza [MS]
33 REAL $SOFTN_END[12] ;Limitare pe axe negativ[I] (I=1:A1,I=7:E1) [MM,GRAD]
34 $SOFTN_END[1]=-185.0
35 $SOFTN_END[2]=-140.0
36 $SOFTN_END[3]=-120.0
37 $SOFTN_END[4]=-350.0
38 $SOFTN_END[5]=-125.0
39 $SOFTN_END[6]=-350.0
40 $SOFTN_END[7]-0.0
41 $SOFTN_END[8]-0.0
42 $SOFTN_END[9]-0.0
43 $SOFTN_END[10]-0.0
44 REAL $SOFTP_END[12] ;Limitare pozitiva pe axa[I] (I=1:A1,I=7:E1) [MM,GRAD]
45 $SOFTP_END[1]=-185.0
46 $SOFTP_END[2]=-5.0

```

Fig.12.Realizarea definirii configuratiei robotului.

De asemenea pentru realizarea programului se vor declara semnalele pentru primul robot cat si pentru cel de-al doilea pentru a realiza programarea.

```

1 DEFDAT masina
2 ;FOLD EXTERNAL DECLARATIONS:$(PE)%MKUKATPBASIS,%CEXT,%VCOMMON,%P
3 ;FOLD BASISTECH EXT:$(PE)%MKUKATPBASIS,%CEXT,%VEXT,%P
4 EXT BAS (BAS_COMMAND :IN,REAL :IN )
5
6 ;ENDFOLD (BASISTECH EXT)
7 ;FOLD USER EXT:$(E)%MKUKATPUSER,%CEXT,%VEXT,%P
8 ;Make your modifications here
9
10 ;ENDFOLD (USER EXT)
11 ;ENDFOLD (EXTERNAL DECLARATIONS)
12 ENDDAT
13 &ACCESS RV
14 &PARAM DISKPATH = MDA
15 &PARAM VERSION = 1.0.0
16 &REL 1
17 DEFDAT $masina PUBLIC
18 CHAR $V_STEUMDA[32]
19 $V_STEUMDA[]="V1.0.0/KUKA8.2" ;versiune
20 SIGNAL $J $IN[1] ;TRUE = Oprea paleta in dreptul lui R1
21 SIGNAL $K $IN[2] ;TRUE = R1 ia piesa de pe paleta
22 SIGNAL $L $IN[3] ;TRUE = Plecarea paletii de la R1 la R2
23 SIGNAL $M $IN[4] ;TRUE = Masurarea si debavurarea piesei
24 SIGNAL $A $OUT[1] ;TRUE = OUT Oprea paleta in dreptul lui R1 a fost realizata
25 SIGNAL $B $OUT[2] ;TRUE = OUT R1 a luat piesa de pe paleta
26 SIGNAL $C $OUT[3] ;TRUE = OUT Plecarea paletii de la R1 la R2 se incheie
27 SIGNAL $D $OUT[4] ;TRUE = OUT Masurarea si debavurarea piesei
28 SIGNAL $I $IN_HOME ; Robot in pozitia de acasa
29 SIGNAL $TIMER1 $TIMER[1] ;Timpul de oprire paleta in dreptul lui R1 si asteptare preluare piesa
30 SIGNAL $TIMER2 $TIMER[2] ;Timpul de masurare si debavurare a piesei
31 SIGNAL $S $FLAG[1] ; Raspuns oprea paleta in dreptul lui R1
32 SIGNAL $SS $FLAG[2] ; Raspuns R1 ia piesa de pe paleta
33 SIGNAL $T $FLAG[3] ; Raspuns plecarea paletii de la R1 la R2
34 SIGNAL $TT $FLAG[4] ; Raspuns masurarea si debavurarea piesei
35

```

Fig.13.Declarare semnale pentru primul robot

```

1 DEFDAT masina
2 ;FOLD EXTERNAL DECLARATIONS:$(PE)%MKUKATPBASIS,%CEXT,%VCOMMON,%P
3 ;FOLD BASISTECH EXT:$(PE)%MKUKATPBASIS,%CEXT,%VEXT,%P
4 EXT BAS (BAS_COMMAND :IN,REAL :IN )
5
6 ;ENDFOLD (BASISTECH EXT)
7 ;FOLD USER EXT:$(E)%MKUKATPUSER,%CEXT,%VEXT,%P
8 ;Make your modifications here
9
10 ;ENDFOLD (USER EXT)
11 ;ENDFOLD (EXTERNAL DECLARATIONS)
12 ENDDAT
13 &ACCESS RV
14 &PARAM DISKPATH = MDA
15 &PARAM VERSION = 1.0.0
16 &REL 1
17 DEFDAT $masina PUBLIC
18 CHAR $V_STEUMDA[32]
19 $V_STEUMDA[]="V1.0.0/KUKA8.2" ;versiune
20 SIGNAL $N $IN[1] ;TRUE = Detectarea piesei de pe suportul intermediar
21 SIGNAL $O $IN[2] ;TRUE = R2 ia piesa de pe suportul intermediar
22 SIGNAL $P $IN[3] ;TRUE = Inlaturare bavura cu jet de aer
23 SIGNAL $Q $IN[4] ;TRUE = R2 pune piesa pe paleta
24 SIGNAL $R $IN[5] ;TRUE = Paleta pleaca
25 SIGNAL $S $OUT[1] ;TRUE = OUTDetectarea piesei de pe suportul intermediar
26 SIGNAL $F $OUT[2] ;TRUE = OUTR2 a luat piesa de pe suportul intermediar
27 SIGNAL $G $OUT[3] ;TRUE = OUTInlaturare bavurii cu jet de aer se incheie
28 SIGNAL $H $OUT[4] ;TRUE = OUTR2 a pus piesa pe paleta
29 SIGNAL $I $OUT[5] ;TRUE = OUTPaleta a pleacat
30 SIGNAL $I $IN_HOME ; Robot in pozitia de acasa
31 SIGNAL $TIMER1 $TIMER[1] ;Timpul de inlaturare a bavurii cu jet de aer
32 SIGNAL $TIMER2 $TIMER[2] ;Timpul de oprire a paletii in dreptul lui R2
33 SIGNAL $U $FLAG[1] ; Raspuns detectarea piesei de pe suportul intermediar
34 SIGNAL $V $FLAG[2] ; Raspuns R2 ia piesa de pe suportul intermediar
35 SIGNAL $W $FLAG[3] ; Raspuns inlaturare bavura cu jet de aer
36 SIGNAL $X $FLAG[4] ; Raspuns R2 pune piesa pe paleta
37 SIGNAL $Y $FLAG[5] ; Raspuns plecarea paleta
38

```

Fig.14.Declarare semnale pentru cel de-al doilea robot

In continuare am sa prezint capturi din programul Visual Work in care a fost realizata programarea pe semnale pentru cei doi roboti. Pentru primul robot au fost realizata urmatoarea programare pe semnale.

```

1  &ACCESS RVF
2  &COMMENT HANDLER on external automatic
3  ;Oprirea paletei in dreptul R1
4  DEF (INSTR:IN, INSTR:OUT, INSTR:FLAG )
5  ;Cand paleta se opreste in dreptul lui R1 ea devine adevarata atunci
6  ;Oprirea paletei R1 devine adevarat deci paleta se opreste atunci si robotul
7  ;poate lua piesa de pe paleta.
8  ;Cand oprirea paletei se realizeaza iar raspunsul de oprire a paletei este pozitiv.
9
10 IF ($OUT[1]<>0)
11 THEN
12   ($FLAG[1]=TRUE AND $FLAG[2]=TRUE)
13 ELSE
14   ($FLAG[1]=TRUE THEN $IN[1]=TRUE)
15 ENDIF
16 ;Cand paleta pleaca din dreptul lui R1 ea devine 0 atunci oprirea paletei devine falsa
17 ;deci paleta pleaca atunci robotul nu mai are ce piesa sa ia.
18 ;In consecinta oprirea paletei nu se mai realizeaza ea pleaca mai departe
19 ;iar raspunsul de oprire a paletei este negativ.
20
21
22 IF ($OUT[1]==0)
23 THEN
24   ($FLAG[1]=FALSE AND $FLAG[2]=FALSE)
25 ELSE
26   ($FLAG[1]=FALSE THEN $IN[1]=FALSE)
27 ENDIF
28 END

```

Fig.15. Programarea opririi paletei in dreptul lui R1

```

31 ;R1 ia piesa de pe paleta
32 DEF (INSTR:IN, INSTR:OUT, INSTR:FLAG )
33 ;Cand primul robot ia piesa de pe paleta atunci robotul ia piesa de pe paleta
34 ;atunci cand paleta este oprita.
35 ;In consecinta primul robot ia piesa de pe paleta iar raspunsul robotului de luare a piesei
36 ;de pe palet este pozitiv.
37
38 IF ($OUT[2]<>0)
39 THEN
40   ($FLAG[2]=TRUE AND $FLAG[1]=TRUE)
41 ELSE
42   ($FLAG[2]=TRUE THEN $IN[2]=TRUE)
43 ENDIF
44 ;Cand primul robot nu ia piesa de pe paleta valoarea devine 0 rezultand ca robotul
45 ;nu ia piesa de pe paleta atunci cand banda merge si paleta este pornita.
46 ;In consecinta primul robot nu ia piesa de pe paleta iar raspunsul robotului de luare a piesei
47 ;de pe palet este negativ.
48
49 IF ($OUT[2]==0)
50 THEN
51   ($FLAG[2]=FALSE AND $FLAG[1]=FALSE)
52 ELSE
53   ($FLAG[2]=FALSE THEN $IN[2]=FALSE)
54 ENDIF
55 END

```

Fig.16. Programarea R1 ia piesa de pe paleta

```

83 ;Masurarea si debavurarea piesei pe dispozitivul special
84 DEF (INSTR:IN, INSTR:OUT, INSTR:FLAG )
85 ;Cand se realizeaza masurarea piesei si debavurarea acesteia atunci
86 ;semnalul statiei de masurare a piesei si debavurare a devine pozitiv
87 ;si paleta dintre robotul 1 si robotul 2 poate pleca.
88 ;In consecinta se realizeaza masurarea si debavurarea piesei
89 ;iar raspunsul la masurare si debavurare a piesei este pozitiv.
90
91 IF ($OUT[4]<>0)
92 THEN
93   ($FLAG[4]=TRUE AND $FLAG[3]=TRUE)
94 ELSE
95   ($FLAG[4]=TRUE THEN $IN[4]=TRUE)
96 ENDIF
97 ;Cand nu se realizeaza masurarea piesei si debavurarea acesteia atunci semnalul statiei
98 ;de masurare a piesei si debavurare a devine negativ si paleta dintre robotul 1 si robotul 2
99 ;nu poate pleca deoarece este incarcat cu noua piesa pentru debavurat.
100 ;In consecinta daca nu se realizeaza masurarea si debavurarea piesei raspunsul la masurare
101 ;si debavurare a piesei este negativ.
102
103
104 IF ($OUT[4]==0)
105 THEN
106   ($FLAG[4]=FALSE AND $FLAG[3]=FALSE)
107 ELSE
108   ($FLAG[4]=FALSE THEN $IN[4]=FALSE)
109 ENDIF
110 END

```

Fig.18. Programarea masurarii si debavurarii piesei pe dispozitivul special

Pentru cel de-al doilea robot a fost realizata urmatoarea programare pe semnale:

```

1  &ACCESS R
2  &PARAM KUKATGRP_VERSION=8.0.0
3  &COMMENT Gripper Routines
4  ;Detectarea piesei cu ajutorul senzorilor pe suportul intermediar
5  DEF (INSTR:IN, INSTR:OUT, INSTR:FLAG )
6  ;Cand se realizeaza detectarea piesei pe suportul intermediar,
7  ;atunci semnalul de detectare a piesei pe suport este pozitiv
8  ;iar cel de-al doilea robot poate lua piesa de pe suport pentru a efectua urmatoarea operatie.
9  ;Daca este detectata piesa pe suportul intermediar atunci
10 ;raspunsul la detectare a piesei pe suport este pozitiv.
11
12 IF ($OUT[1]<>0)
13 THEN
14   ($FLAG[1]=TRUE AND $FLAG[2]=TRUE)
15 ELSE
16   ($FLAG[1]=TRUE THEN $IN[1]=TRUE)
17 ENDIF
18 ;Cand nu se realizeaza detectarea piesei pe suportul intermediar,
19 ;atunci semnalul de detectare a piesei pe suport este negativ
20 ;iar cel de-al doilea robot nu poate lua piesa de pe suport pentru a efectua urmatoarea operatie.
21 ;Cand nu este detectata piesa pe suportul intermediar
22 ;atunci raspunsul la detectare a piesei pe suport este negativ
23
24 IF ($OUT[1]==0)
25 THEN
26   ($FLAG[1]=FALSE AND $FLAG[2]=FALSE)
27 ELSE
28   ($FLAG[1]=FALSE THEN $IN[1]=FALSE)
29 ENDIF
30 END

```

Fig.19.Programarea detectarii piesei cu ajutorul senzorilor pe suportul intermediar

```

61 ;Inlaturarea bavurii cu jet de aer
62 DEF (INSTR:IN, INSTR:OUT, INSTR:FLAG )
63 ;Cand se realizeaza indepartarea bavurii atunci semnalul de inlaturare a bavurii este pozitiv
64 ;iar robotul nu mai poate lua nici o piesa.
65 ;Cand indepartarea bavurii se realizeaza iar semnalul pentru indepartare a bavurii este pozitiv.
66
67
68 IF ($OUT[3]<>0)
69 THEN
70   ($FLAG[3]=TRUE AND $FLAG[2]=FALSE)
71 ELSE
72   ($FLAG[3]=TRUE THEN $IN[3]=TRUE)
73 ENDIF
74 ;Cand nu se realizeaza indepartarea bavurii atunci semnalul de inlaturare a bavurii este negativ
75 ;iar robotul poate lua piesa de pe suport.
76 ;Cand indepartarea bavurii nu se realizeaza atunci semnalul pentru indepartare a bavurii este negativ.
77
78 IF ($OUT[3]==0)
79 THEN
80   ($FLAG[3]=FALSE AND $FLAG[2]=TRUE)
81 ELSE
82   ($FLAG[3]=FALSE THEN $IN[3]=FALSE)
83 ENDIF
84 END

```

Fig.21.Programarea inlaturarii bavurii cu jet de aer

```

111 ;Paleta pleaca
112 DEF (INSTR:IN, INSTR:OUT, INSTR:FLAG )
113 ;Cand paleta pleaca atunci raspunsul la plecarea paletii este pozitiv
114 ;si cel de-al doilea robot nu poate pune piese pe paleta.
115 ;Cand semnalul de plecare al paletii este adevarat atunci raspunsul la plecarea paletii
116 ;este si el pozitiv deci paleta poate pleca din statie.
117
118
119 IF ($OUT[5]<>0)
120 THEN
121   ($FLAG[5]=TRUE AND $FLAG[4]=FALSE)
122 ELSE
123   ($FLAG[5]=TRUE THEN $IN[5]=TRUE)
124 ENDIF
125 ;Cand paleta nu pleaca atunci raspunsul la plecarea paletii este negativ
126 ;si cel de-al doilea robot poate pune piese pe paleta
127 ;Cand semnalul de plecare al paletii este fals atunci raspunsul la plecarea paletii
128 ;este si el negativ deci paleta nu poate pleca din statie.
129
130 IF ($OUT[5]==0)
131 THEN
132   ($FLAG[5]=FALSE AND $FLAG[4]=TRUE)
133 ELSE
134   ($FLAG[5]=FALSE THEN $IN[5]=FALSE)
135 ENDIF
136 END

```

Fig.23.Pogramarea plecatii paletii de pe banda

In continuare se va prezenta si realizarea simularii in mediul de lucru KukaSim Pro . Pentru a putea realiza simularea a fost nevoie de invatarea punctelor pentru fiecare obiect.In imaginile de mai jos se va detalia meniul de invatare a punctelor.Se va lua exemplu una din operatiile realizate de robot.Pentru a se putea repeta aceasta operatie se realizeaza mai intai un program principal care acceseaza la randul lui alte programe in cazul nostru inca trei programe.Aceasta reprezinta rutina/bucloa /numarul de operatii pe care robotul il desfasoara.Pentru a exemplifica robotul in cazul acesta va face aceeasi operatie de trei ori in care realizeaza un ciclu repetitiv.

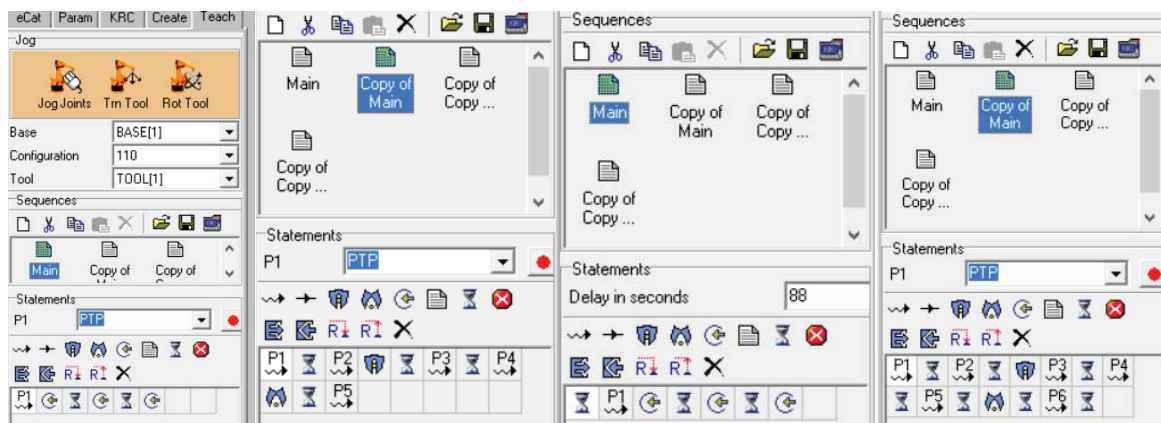


Fig.24.Prezentare a unui exemplu de realizare a unei operatii realizate de unul din robotii aflatii in celula

Dupa realizarea acestor operatii atat pentru roboti cat si pentru conveior si celelalte elemente sa realizat filmul de prezentare.

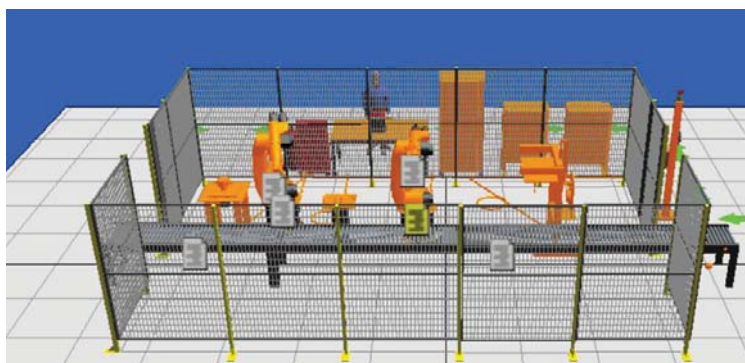


Fig.25.Captura din programul Kuka Sim Pro 1 cu realizarea celulei robotizate studiate.

3. Concluzii

In aceasta lucrare au fost aduse ca si contributii originale realizarea programarii offline a aplicatiei robotizate pentru debavurare. In celula robotizata se realizeaza debavurarea unui arbore secundar al unei cutii de viteze. Este prezentata organizarea logica a evenimentelor si definirea semnalelor. Aceasta se realizeaza in programul Work Visual 3.1. Este prezentata de asemenea simularea offline a functionarii pe baza de semnale a aplicatiei. Simularea este realizata in programul Kuka Sim 1.1.

Bibliografie.

1. [1]- Nicolescu A. – Proiectarea Robotilor Industriali. Partea I. Conceptul sistemic unitar de robot integrat în mediul tehnologic. Subsistemul mecanic al RI. Motoare de actionare utilizate la RI, UPB, 1997
2. [2]- Nicolescu, A., Roboti Industriali – Vol.1 Sub sisteme si ansambluri componente. Structura axelor comandate numeric ale RI, ISBN 973 – 30 – 1244 – 0, Editura Didactica si Pedagogica RA, 2005, Bucuresti
3. [3]- Nicolescu, A. – Implementarea Robotilor Industriali in Sistemele de Productie, note de curs si metodologii de proiectare, UPB, 2018