

SPRAWOZDANIE 2

Manipulator

Data wykonania:	26.11.2021	Nazwisko i imię, numer albumu
Numer grupy:	6	Pieprzycki Kamil, 402037

1. Równoważne tablice stanów opisujące działanie napędów NX i NY:

START_NX	START_NY	NX	NY
0	0	0	0
0	1	0	1
1	0	1	0
1	1	0	0

BX00vBX01vBX02	BY0	BY1	NX	NY
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
0	1	1	-	-
1	0	0	-	-
1	0	1	1	0
1	1	0	1	1
1	1	1	-	-

2. Kod programu:

- Zmienne programu głównego:

```
1 PROGRAM PLC_PRG
2 VAR
3     S1,S2,S3,S4:BOOL; //zmienne przyciskow S1-start S2-stop S3-kontrola chwytaka, S4- tryb automatyczny
4     BX00,BX01,BX02:BOOL; //zmienne okreslajace stany stabilne na osi X
5     BY00,BY01:BOOL; //zmienne okreslajace stany stabilne na osi Y
6     B0,B1,B2,B3:BOOL;
7     zbocz_impl100ms : R_TRIG;
8     impulsy:BOOL;
9
10    zmiennaINT: INT;
11 //zmienne funkcji uruchamianie:
12    U1: uruchamianie;
13    war_ruchu_X:BOOL;
14    war_ruchu_Y:BOOL;
15 //zmienne funkcji manipulator:
16    DIR_X_KP, START_X_KP:BOOL; //zmienne okreslajace kierunek i stan napedu na osi X
17    DIR_Y_KP, START_Y_KP:BOOL; //zmienne ruch OY
18    GOTO:INT;
19    M1: manipulator;
20    polozenie_x: INT;
21    polozenie_y: INT;
22    Start: BOOL;
23    CHWYTAK:BOOL;
24    LICZBA_CYKLI:INT;
25 //funkcja stan
26    St1: stan;
27    stan_manipulatora:INT;
28 END_VAR
```

- Program główny

```
1 zbocz_impl100ms( CLK := GVL.impl100ms, Q=>impulsy);
2 IF impulsy THEN
3     zmiennaINT:= zmiennaINT+1;
4 END_IF
5
6 //wywołanie funkcji implementującej przycisk stop do manipulatora
7 U1(START_X_KP:=START_X_KP, _START_Y_KP := START_Y_KP, _STOP_KP :=S2, _war_ruchu_X => war_ruchu_X, _war_ruchu_Y => war_ruchu_Y);
8 //wywołanie bloku funkcyjnego sterującego działaniem manipulatora
9 M1(impuls:=impulsy, _startManipulatora:=Start, _DIR_X_KP:=DIR_X_KP, _START_X_KP:=war_ruchu_X, _DIR_Y_KP:=DIR_Y_KP, _START_Y_KP:=war_ruchu_Y,
10 _GOTO_KP:=GOTO, _CHWYTAK_KP:=CHWYTAK, _S1_KP:=S1, _S2_KP:=S2, _S3_KP:=S3, _S4_KP:=S4, _BX00_KP=>BX00, _BX01_KP=>BX01, _BX02_KP=>BX02, _BY00_KP=>BY00, _BY01_KP=>BY01,
11 out_pozycjaX=>polozenie_x, out_pozycjaY=>polozenie_y, _LICZBA_CYKLI_KP:=LICZBA_CYKLI);
12 //wywołanie bloku funkcyjnego zwracającego stan manipulatora
13 St1(_Bx00_KP:=BX00, _Bx01_KP:=BX01, _Bx02_KP:=BX02, _By0_KP:=BY00, _By1_KP:=BY01, _stan_manipulatora => stan_manipulatora);
14
15
```

- Zmienne bloku funkcyjnego „manipulator”:

```

1 FUNCTION BLOCK manipulator
2 VAR_INPUT
3   _impuls:BOOL;
4   _startManipulatora:BOOL;
5
6
7 END_VAR
8 VAR_IN_OUT
9   _CHWYTAK_KP:BOOL;
10  _DIR_X_KP, _START_X_KP:BOOL; //zmienne okreslajace kierunek i stan napedu na osiX
11  _DIR_Y_KP, _START_Y_KP:BOOL;
12  _S1_KP, _S2_KP, _S3_KP, _S4_KP:BOOL; //zmienna forsujaca stan pocz
13  _GOTO_KP:INT;
14 END_VAR
15 VAR_OUTPUT
16
17   _BX00_KP, _BX01_KP, _BX02_KP, _BY00_KP, _BY01_KP:BOOL; //zmienne czujnikow
18   out_pozycjaX:INT;
19   out_pozycjaY:INT;
20   _LICZBA_CYKLI_KP:INT;
21 END_VAR
22 VAR_STAT
23   _znacznikzb_pocz_KP:BOOL;
24   _reset_stopu_KP: F_TRIG;
25   _reset_automatuy : R_TRIG;
26   _reset_stopu_pom_KP:BOOL;
27   _Stop_KP:BOOL;
28   _pozycjaX_KP:INT:=10;
29   _pozycjaY_KP:INT:=10;
30   _zerowanie_KP:BOOL;
31   _CHWYTAK_POM_KP:BOOL;
32   _ALGORYTM_KP:INT;
33   _T1_KP:TON;
34   _czas_KP:TIME := T#5S;
35   _czekaj_KP:BOOL;
36   _WYBOR_RUCHU_KP:INT;
37
38 END_VAR
39

```

- Program symulujący działanie manipulatora, obsługujący automatyczny algorytm i zwracający wartości bitów DIR i START, symulujący działania chwytaka i czujników BX i BY

```

1 //ustawianie zmiennej pomocniczej _Stop ktora jesli bedzie w stanie niskim uniemozli wyzerowania manipulatora
2 _Stop_KP := _Stop_KP OR _S2_KP; //cewka set gdy nacisniemy przycisk S2
3
4 //reset zmiennej _Stop gdy zostanie wykryte zboczne opadajace na zmiennej _zerowanie
5 _Stop_KP := _Stop_KP AND NOT _reset_stopu_pom_KP;
6 _reset_stopu_KP(CLK := _zerowanie_KP, Q=> _reset_stopu_pom_KP);
7
8 //kod algorytmu zerowania manipulatora uruchamia sie gdy zostanie wcisniety przycisk _S1 i manipulator jest wylaczony
9 _zerowanie_KP := _zerowanie_KP OR (_S1_KP AND _Stop_KP);
10 IF _zerowanie_KP AND NOT _BY00_KP THEN //zerowanie manipulatora na osi Y
11   _pozycjaY_KP := _pozycjaY_KP + 2 ;
12   out_pozycjaY := _pozycjaY_KP;
13 ELSIF _zerowanie_KP AND NOT _BX00_KP AND _BY00_KP THEN //zerowanie manipulatora na osi X
14   _pozycjaX_KP := _pozycjaX_KP - 2 ;
15   out_pozycjaX := _pozycjaX_KP;
16 END_IF
17 IF _BX00_KP AND _BY00_KP THEN //gdy chwytak w pozycji poczatkowej -> zerowanie w stanie niskim
18   _zerowanie_KP := _S1_KP;
19   _GOTO_KP:=0;
20 END_IF
21 IF NOT _Stop_KP THEN
22 CASE _WYBOR_RUCHU_KP OF // case zawierajacy przypadki ruchu i zwracajacy odpowiednie wartosci bitow DIR i START
23 0:
24   //spoczynek
25   _DIR_X_KP:= FALSE;
26   _START_X_KP:= FALSE; //zmienne okreslajace kierunek i stan napedu na osiX
27   _DIR_Y_KP:=FALSE;
28   _START_Y_KP:=FALSE;
29 1:
30   //ruch w prawo
31   _DIR_X_KP:= TRUE;
32   _START_X_KP:= TRUE;
33   _DIR_Y_KP:=FALSE;
34   _START_Y_KP:=FALSE;
35   _pozycjaX_KP := _pozycjaX_KP + 2 ;
36   out_pozycjaX := _pozycjaX_KP;
37 2:
38   //ruch w lewo
39   _DIR_X_KP:= FALSE;
40   _START_X_KP:= TRUE;
41   _DIR_Y_KP:=FALSE;
42   _START_Y_KP:=FALSE;
43   _pozycjaX_KP := _pozycjaX_KP - 2 ;
44   out_pozycjaX := _pozycjaX_KP;
45 3:
46   //ruch w gore
47   _DIR_X_KP:= FALSE;
48   _START_X_KP:= FALSE;
49   _DIR_Y_KP:=TRUE;
50   _START_Y_KP:=TRUE;
51   _pozycjaY_KP := _pozycjaY_KP - 2 ;
52   out_pozycjaY := _pozycjaY_KP;
53 4:
54   //ruch w dol
55   _DIR_X_KP:= FALSE;
56   _START_X_KP:= FALSE;
57   _DIR_Y_KP:=FALSE;
58   _START_Y_KP:=TRUE;
59   _pozycjaY_KP := _pozycjaY_KP + 2 ;
60   out_pozycjaY := _pozycjaY_KP;
61
62 END CASE
63 ELSIF _Stop_KP THEN // gdy stop wcisniety
64   _DIR_X_KP:= FALSE;
65   _START_X_KP:= FALSE; //zmienne okreslajace kierunek i stan napedu na osiX
66   _DIR_Y_KP:=FALSE;
67   _START_Y_KP:=FALSE;
68
69 END_IF

```

```

71 //zadeklarowanie czujnikow
72 _BX00_KP := _pozycjaX_KP =10;
73 _BX01_KP := _pozycjaX_KP =90;
74 _BX02_KP := _pozycjaX_KP =180;
75 _BY00_KP := _pozycjaY_KP=10;
76 _BY01_KP := _pozycjaY_KP=-160;
77
78 //algorytm ruchu manipulatora
79 IF NOT _Stop_KP AND NOT _zerowanie_KP THEN //jezeli stop i nie wyzerujemy manipulatora za pomoca S1 to brak ruchu
80 CASE _GOTO_KP OF
81 0: //ruch do pozycji 0 najpierw BY0 potem BX0
82 IF NOT _BY00_KP AND (_BX00_KP OR _BX01_KP OR _BX02_KP) THEN//ruch w dol
83 _WYBOR_RUCHU_KP:=4;
84 END_IF
85 IF NOT _BX00_KP AND _BY00_KP THEN
86 _WYBOR_RUCHU_KP:=2;
87 END_IF
88 IF _BX00_KP AND _BY00_KP THEN
89 _WYBOR_RUCHU_KP:=0;
90 END_IF
91 1: //ruch do pozycji 1 najpierw os BTO potem BX0 potem BY1
92 IF NOT _BY00_KP AND (_BX00_KP OR _BX01_KP OR _BX02_KP) THEN//ruch w dol
93 _WYBOR_RUCHU_KP:=4;
94 END_IF
95 IF NOT _BX00_KP AND _BY00_KP THEN
96 _WYBOR_RUCHU_KP:=2;
97 END_IF
98 IF NOT _BY01_KP AND _BX00_KP THEN
99 _WYBOR_RUCHU_KP:=3;
100 END_IF
101 IF _BX00_KP AND _BY01_KP THEN
102 _WYBOR_RUCHU_KP:=0;
103 END_IF
104 2: //ruch do pozycji 2 najpierw BY0 potem OX1
105 IF NOT _BY00_KP AND (_BX00_KP OR _BX01_KP OR _BX02_KP) THEN //jezeli powyzej od pozycji 2 to ruch w dol
106 _WYBOR_RUCHU_KP:=4;
107 END_IF
108 IF NOT _BX01_KP AND _BY00_KP AND _pozycjaX_KP > 90 THEN //jezeli po prawo od pozycji 2 to ruch w lewo
109 _WYBOR_RUCHU_KP:=2;
110 END_IF
111 IF NOT _BX01_KP AND _BY00_KP AND _pozycjaX_KP < 90 THEN //jezeli po lewo od pozycji 2 to ruch w prawo
112 _WYBOR_RUCHU_KP:=1;
113 END_IF
114 IF _BX01_KP AND _BY00_KP THEN
115 _WYBOR_RUCHU_KP:=0;
116 IF NOT _BY00_KP AND (_BX00_KP OR _BX01_KP OR _BX02_KP) THEN //jezeli powyzej od pozycji 2 to ruch w dol
117 _WYBOR_RUCHU_KP:=4;
118 END_IF
119 IF NOT _BX01_KP AND _BY00_KP AND _pozycjaX_KP > 90 THEN //jezeli po prawo od pozycji 2 to ruch w lewo
120 _WYBOR_RUCHU_KP:=2;
121 END_IF
122 IF NOT _BX01_KP AND _BY00_KP AND _pozycjaX_KP < 90 THEN //jezeli po lewo od pozycji 2 to ruch w prawo
123 _WYBOR_RUCHU_KP:=1;
124 END_IF
125 IF _BX01_KP AND NOT _BY01_KP THEN
126 _WYBOR_RUCHU_KP:=3;
127 END_IF
128 IF _BX01_KP AND _BY01_KP THEN
129 _WYBOR_RUCHU_KP:=0;
130 END_IF
131 4: //ruch do pozycji 4 najpierw BY0 potem OX2
132 IF NOT _BY00_KP AND (_BX00_KP OR _BX01_KP OR _BX02_KP) THEN
133 _WYBOR_RUCHU_KP:=4;
134 END_IF
135 IF NOT _BX02_KP AND _BY00_KP THEN
136 _WYBOR_RUCHU_KP:=1;
137 END_IF
138 IF _BX02_KP AND _BY00_KP THEN
139 _WYBOR_RUCHU_KP:=0;
140 END_IF
141 5: //ruch do pozycji 5 najpierw do BY0 potem OX2 potem OY1
142 IF NOT _BY00_KP AND (_BX00_KP OR _BX01_KP OR _BX02_KP) THEN
143 _WYBOR_RUCHU_KP:=4;
144 END_IF
145 IF NOT _BX02_KP AND _BY00_KP THEN
146 _WYBOR_RUCHU_KP:=1;
147 END_IF
148 IF _BX02_KP AND NOT _BY01_KP THEN
149 _WYBOR_RUCHU_KP:=3;
150 END_IF
151 IF _BX02_KP AND _BY01_KP THEN
152 _WYBOR_RUCHU_KP:=0;
153 END_IF
154
155 END_CASE
156 END_IF
157
158 //algorytm chwytaka
159 _CHWYTIK_KP := _CHWYTIK_POM_KP;
160 IF NOT _S4_KP THEN // w trybie autonomicznym odbieramy mozliwosc sterowania chwytakiem za pom przycisku
161 _CHWYTIK_POM_KP:=_S3_KP;
162 END_IF

```

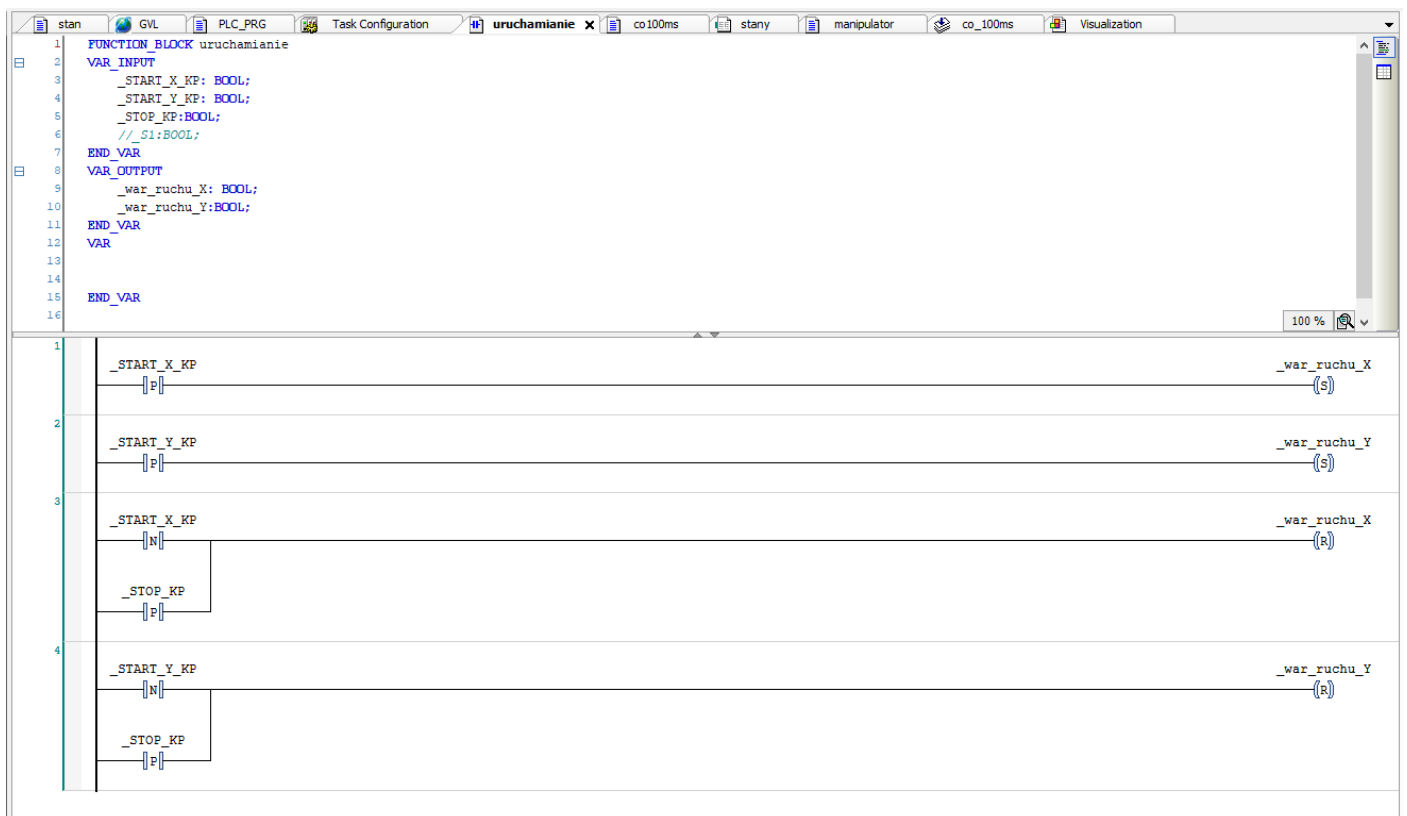
```

170 IF _S1_KP THEN
171   _ALGORYTM_KP:=1; // zerowanie ustawia algorytm autonomicznosci do kroku pierwszego
172 END_IF
173 IF _S4_KP AND NOT _zerowanie_KP THEN
174 CASE _ALGORYTM_KP OF //Manipulator w trybie automatycznym
175 1: //krok1
176   _CHWYTIK_POM_KP:= _CHWYTIK_POM_KP AND NOT (_BX00_KP AND _BY01_KP); //otworz chwytak
177   _GOTO_KP :=3;
178   IF _BY01_KP AND _BX01_KP THEN //jezeli pozycja 3 to nastepny krok
179     _ALGORYTM_KP:=_ALGORYTM_KP+1;
180   END_IF
181 2: //krok 2
182   _CHWYTIK_POM_KP:= _CHWYTIK_POM_KP OR (_BX01_KP AND _BY01_KP); //zamknij chwytak
183   _GOTO_KP :=5;
184   IF _BY01_KP AND _BX02_KP THEN //jezeli pozycja 5 to nastepny krok
185     _ALGORYTM_KP:=_ALGORYTM_KP+1;
186   END_IF
187 3: //krok3
188   _CHWYTIK_POM_KP:= _CHWYTIK_POM_KP AND NOT (_BX02_KP AND _BY01_KP); //otworz chwytak jak pozycja 5
189   _GOTO_KP := 4;
190   _T1_KP(IN:=( _BY00_KP AND _BX02_KP), PT:=_czas_KP, Q => _czekaj_KP); //czekaj 5 sekund gdy pozycja 4
191   IF _BY00_KP AND _BX02_KP AND _czekaj_KP THEN //po odczekaniu w pozycji 4 nastepny krok
192     _ALGORYTM_KP:=_ALGORYTM_KP+1;
193   END_IF
194 4: //krok 4
195   _GOTO_KP :=5;
196   _CHWYTIK_POM_KP:= _CHWYTIK_POM_KP OR (_BX02_KP AND _BY01_KP); //zamknij chwytak
197   IF _BY01_KP AND _BX02_KP AND _CHWYTIK_POM_KP THEN //gdy 5 i chwytak zamkniety to nastepny krok
198     _ALGORYTM_KP:=_ALGORYTM_KP+1;
199   END_IF
200 5: //krok 5
201   _GOTO_KP :=1;
202   _T1_KP(IN:=( _BY01_KP AND _BX00_KP), PT:=_czas_KP, Q => _czekaj_KP); //czekaj 5 sekund w pozycji 1
203   IF _BY01_KP AND _BX00_KP AND _czekaj_KP THEN // po odczekaniu idz do kroku nr 1
204     _ALGORYTM_KP:=1;
205     _LICZBA_CYKLI_KP:= _LICZBA_CYKLI_KP+1; //inkrementacja liczby cykli wykonania algorytmu
206   END_IF
207 END_CASE
208 END_IF
209
210 END_IF

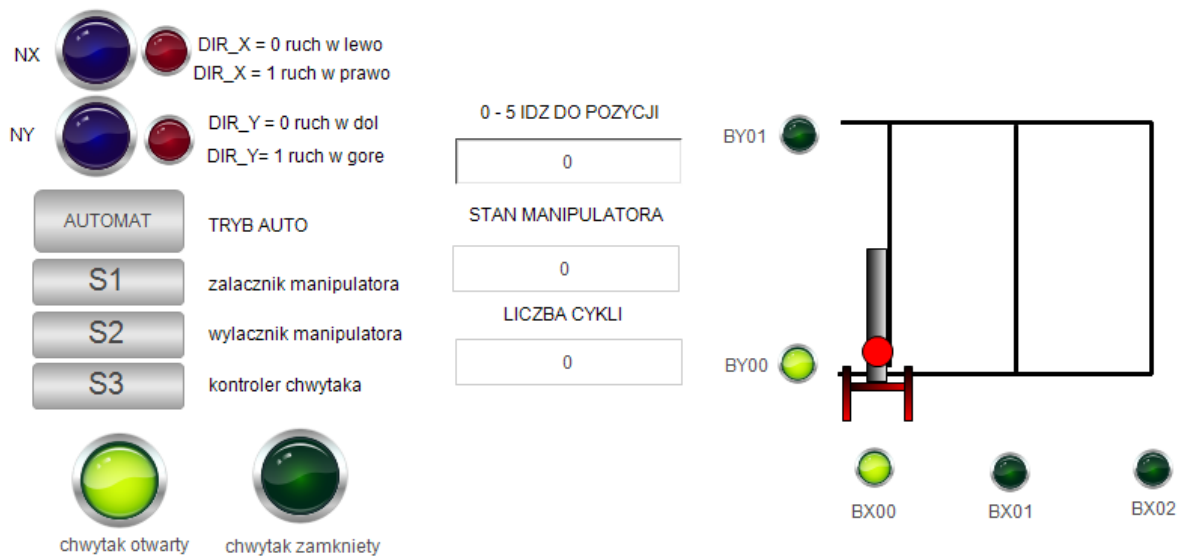
```

100 %

- Blok funkcyjny odpowiadający za obsługę przycisku stop



3. Zrzut ekranu zawierający panel HMI chwytaka :



4. Film prezentujący działanie manipulatora w HMI programu :

https://aghedupl-my.sharepoint.com/:v:/g/personal/pieprzycki_student_agh_edu_pl/EXFAoil4xChCjt5RRp6Yc2oBsrTI7qRDeI13a8Q9TzqMog?e=7VJjeG