2019

Olbricht Automation GmbH

Ağan Şimşek, Orhun Seyhun

[M2S ROBOT MANUAL]

This document contains information on additional features made by integrator on standard robot system

Contents

1. Robot Mastering 3

2. Robot I/O configuration 4

2.1. Master module setup 7

2.2. Analog module setup 7

2.3. I/O cross references 8

2.3.1. List of input and outputs 9

2.4. Robot sensor setup 17

2.5. Robot sensor calibration 18

3. Robot Restore Backup 19

4. Software Structure 20

4.1. Main Program 20

4.2. SPS Program 21

4.3. Paths 22

4.3.1. aa_to170_2 22

4.3.2. ab_to170_2 23

4.3.3. ba_to170_2 23

4.3.4. bb_to170_2 24

4.3.5. find_crate_nt 24

4.3.6. Inspect 26

4.3.7. pick_from150 26

4.3.8. pick_from180 26

4.3.9. pick_scrap 27

4.3.10. place_scrap 27

4.3.11. place_to150 27

4.3.12. place_to180 27

4.3.13. picknplace_to150 27

4.4. Sub Routines 28

4.5. Special Programs 28

4.5.1. Tool_check 28

4.5.2. Sensor Calibration 28

4.5.3. Home 28

4.5.4. Maintenance 28

5. Base 29

6. Tool 31

7. References..33

1. Robot Mastering

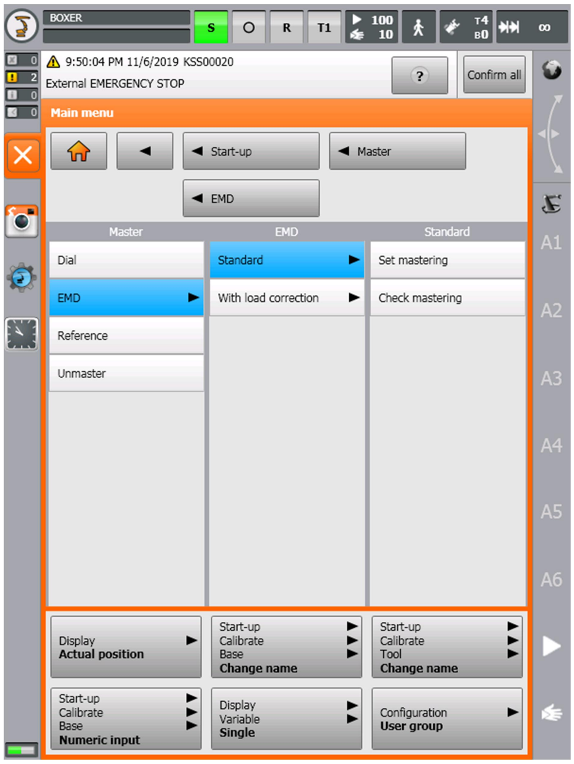


Figure 1 Robot Mastering Type

Robot calibration must be done as in picture; choice standard and should be done without MB .

2. Robot I/O configuration

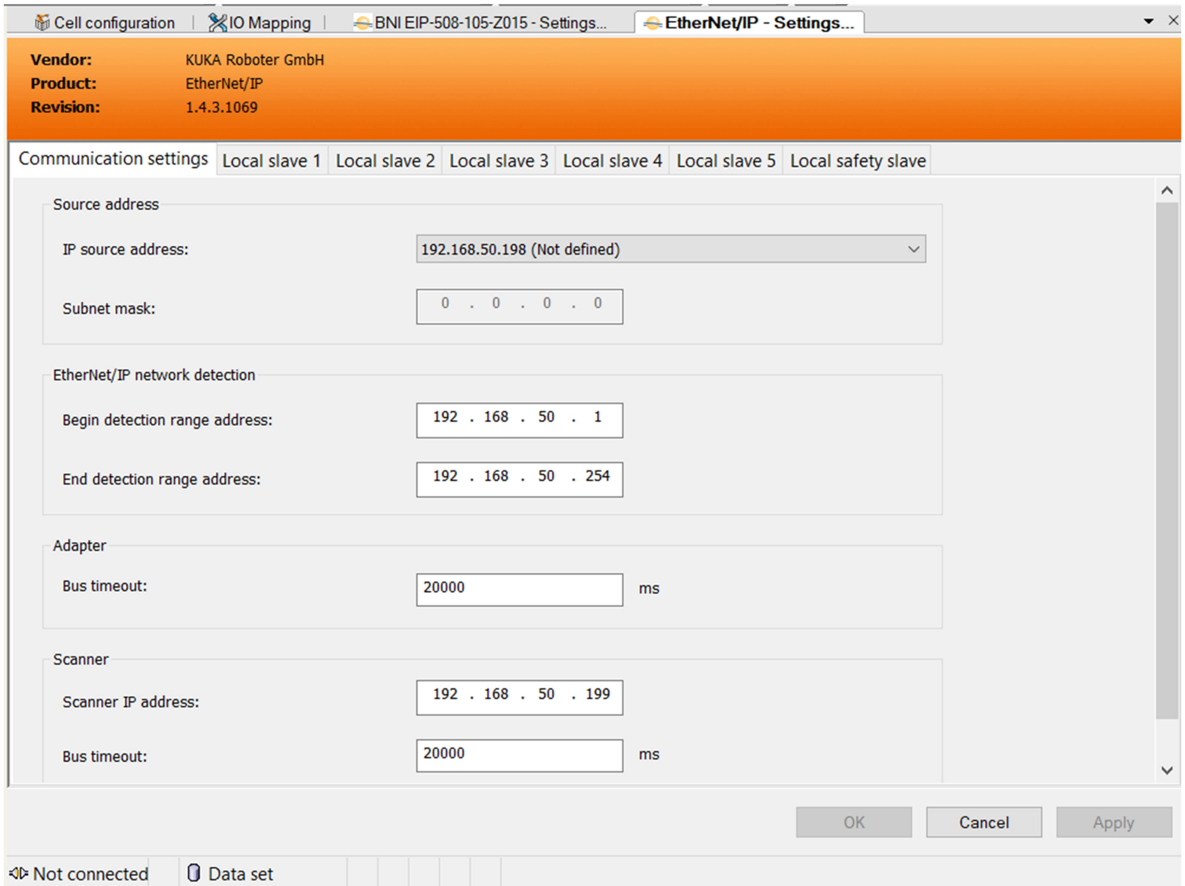


Figure 2 Ethernet IP configuration

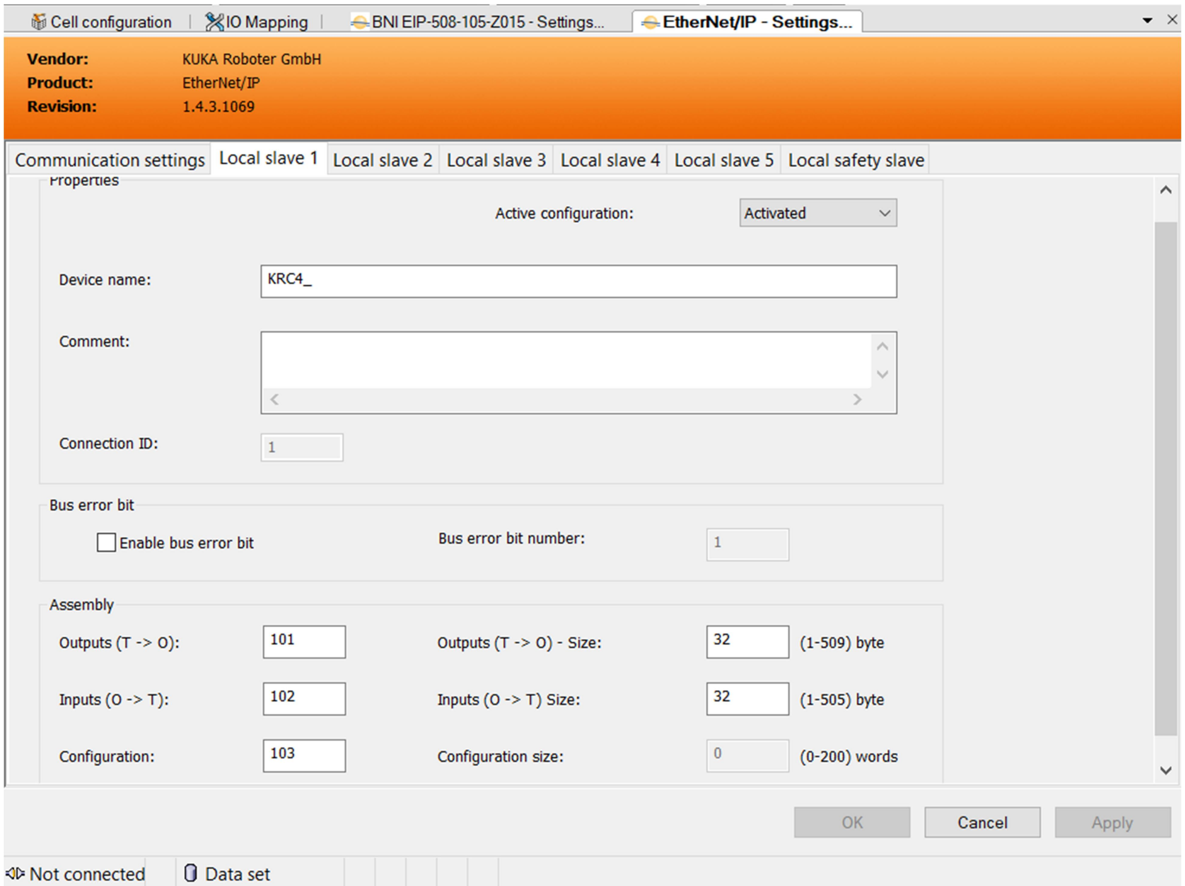


Figure 3 Ethernet IP Slave configuration

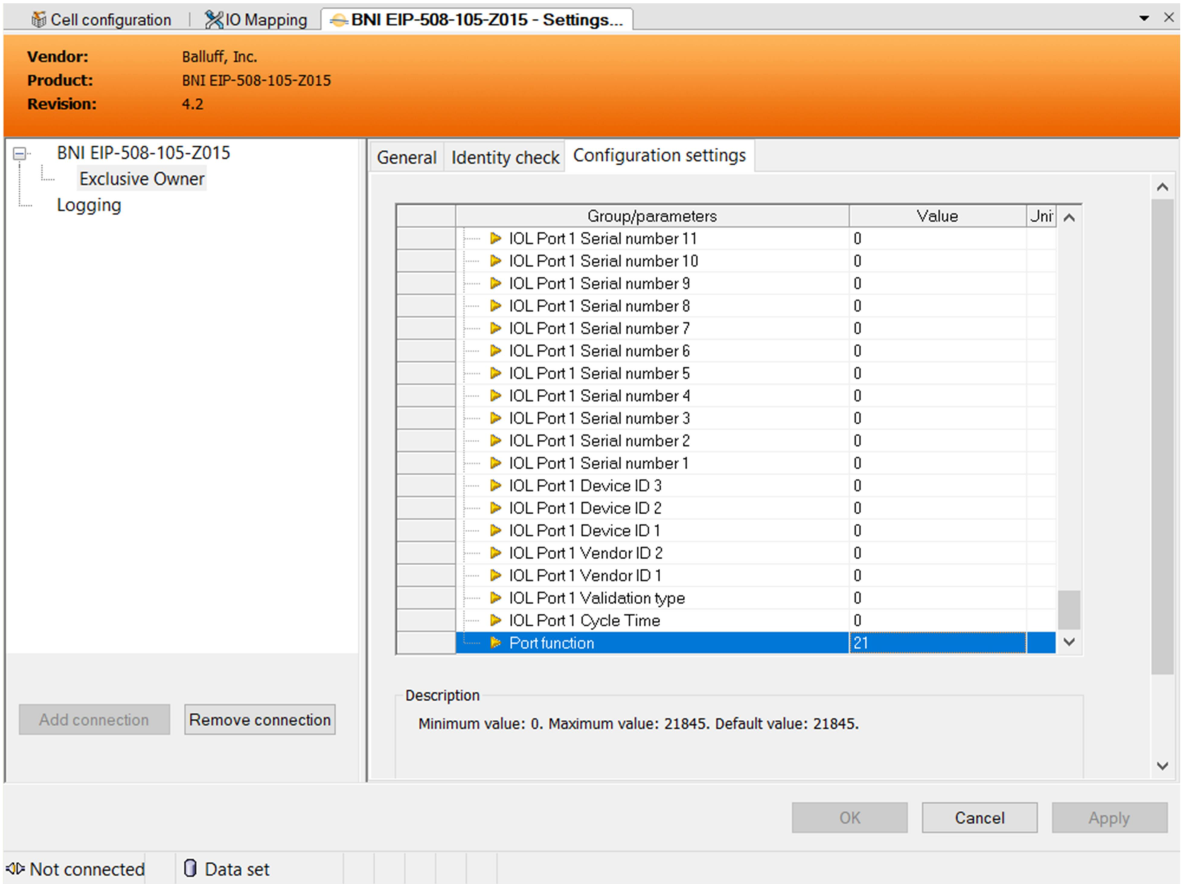


Figure 4 Workvisual I/O Ethernet IP config

2.1. Master module setup

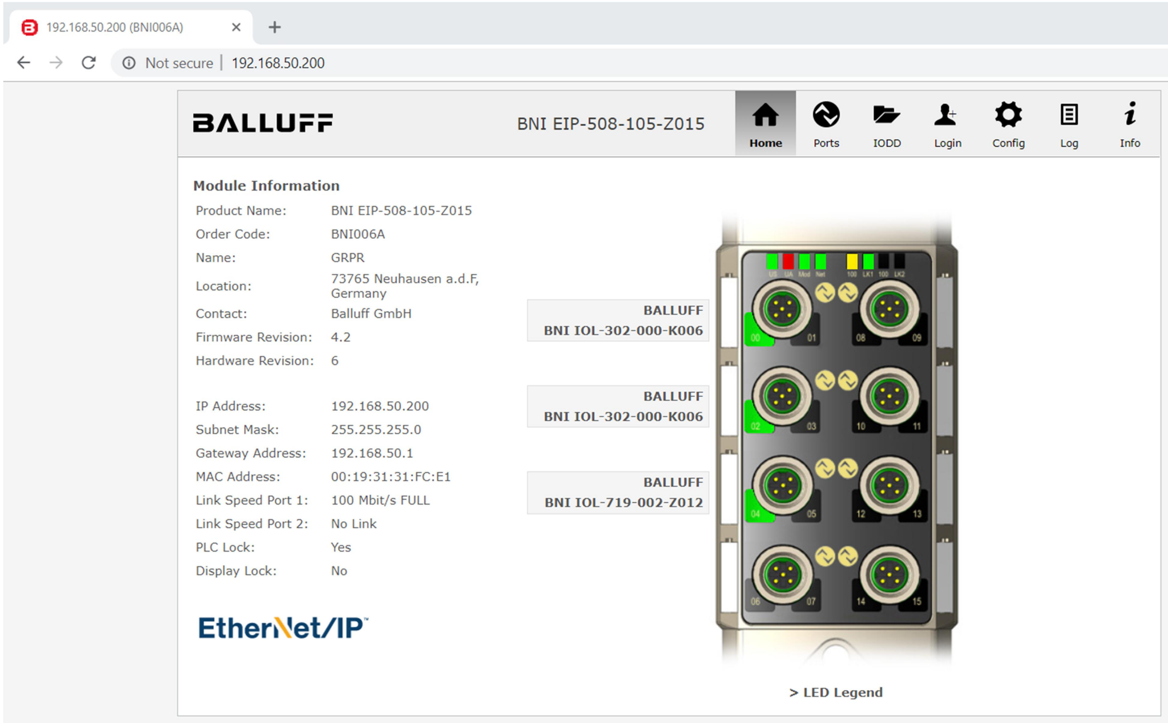


Figure 5 I/O link master model

IP address must be set to 192.168.50.20. Name automatically set by Robot controller if not please set to GRPR.

2.2. Analog module setup

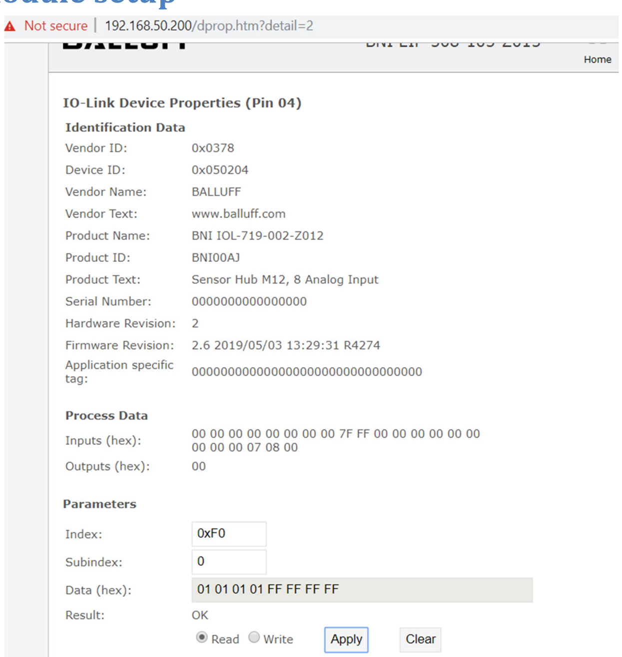


Figure 6 Analog mode

Channel 1 to 4 is use for 4-20mA measuring.5 to 8 not in use.

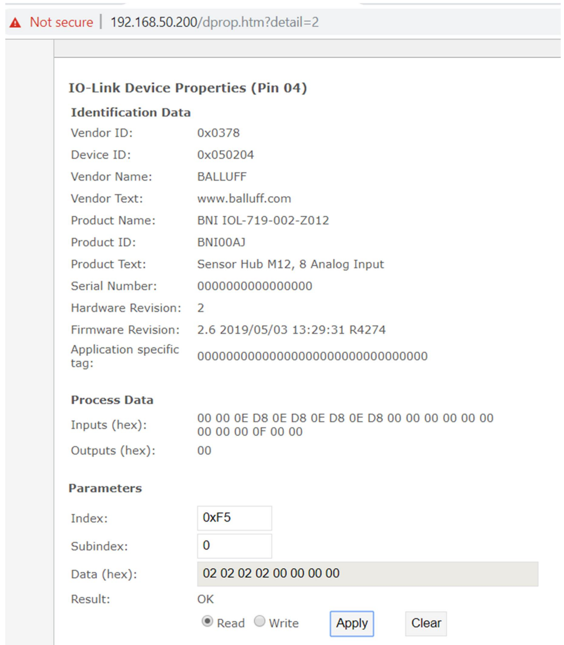


Figure 7 Proses data format

The analog value can be represented as dimensioned in the process data

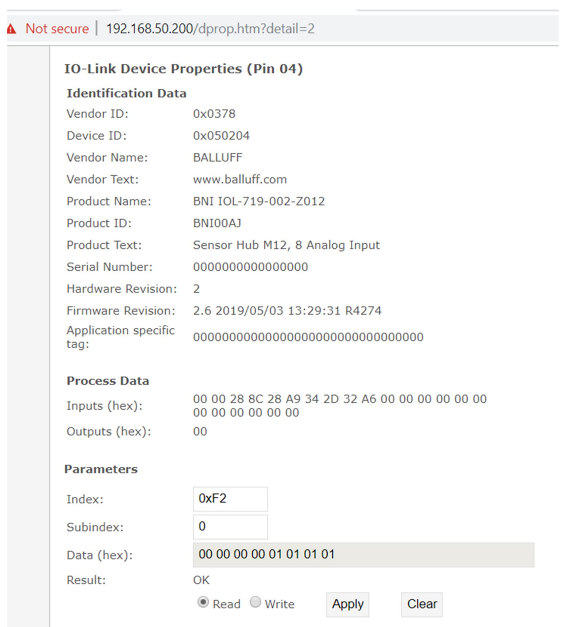


Figure 8 Pin assignment

The source pin 2 selected for analog input.

2.3. I/O cross references

First 32 Bytes are used for communicating with PLC. From 130th byte to 141th ($in[1040]….$in[1135],

$out[1040]…$out[1071]) are used for communicating to I/O modules. $out[1024] $out[1039] I/O link Master module at robot tool.

2.3.1. List of input and outputs

$IN[1]	Prog_no bit0	$OUT[1]	Prog_no ref bit0
$IN[2]	Prog_no bit1	$OUT[2]	Prog_no ref bit1
$IN[3]	Prog_no bit2	$OUT[3]	Prog_no ref bit2
$IN[4]	Prog_no bit3	$OUT[4]	Prog_no ref bit3
$IN[5]	Prog_no bit4	$OUT[5]	Prog_no ref bit4
$IN[6]	Prog_no bit5	$OUT[6]	Prog_no ref bit5
$IN[7]	Prog_no bit6	$OUT[7]	Prog_no ref bit6
$IN[8]	Prog_no bit7	$OUT[8]	Prog_no ref bit7
$IN[9]	Prog_no valid	$OUT[9]	Program active
$IN[10]	Ext start	$OUT[10]	Program no request
$IN[11]	Move enable	$OUT[11]	Application running
$IN[12]	Conf messages	$OUT[12]	Pro move
$IN[13]	Drives off	$OUT[13]	In home
$IN[14]	Drives on	$OUT[14]	Near pos ret
$IN[15]		$OUT[15]	Rob stopped
$IN[16]		$OUT[16]	T1 mode
$IN[17]		$OUT[17]	EXT mode
$IN[18]		$OUT[18]	Robot controller ready
$IN[19]		$OUT[19]	Alarm stop
$IN[20]		$OUT[20]	User safety
$IN[21]		$OUT[21]	Peri ready
$IN[22]		$OUT[22]	Stopmess
$IN[23]		$OUT[23]	Alarm stop
$IN[24]		$OUT[24]	T2 mode
$IN[25]		$OUT[25]	On path
$IN[26]		$OUT[26]	I_O Act conf
$IN[27]		$OUT[27]	Rob calibrated
$IN[28]		$OUT[28]	In home1
$IN[29]		$OUT[29]	robot_hand_stop
$IN[30]	new_crate	$OUT[30]	new_crate_mea_done
$IN[31]	pump running	$OUT[31]	crate mea fail
$IN[32]	simulation	$OUT[32]	tool check fail
$IN[33]	M163 ready	$OUT[33]	SPC READY TO UNPLUG
$IN[34]	M190 ready	$OUT[34]	no grip feed back
$IN[35]	M150 High Pos	$OUT[35]	Safety to move M150
$IN[36]	M150 Mid Pos	$OUT[36]	
$IN[37]	M150 Low Pos	$OUT[37]	
$IN[38]	M180 A Side	$OUT[38]	Safety to move M180
$IN[39]	M180 B Side	$OUT[39]	
$IN[40]	M170 ready	$OUT[40]	
$IN[41]	Authorize to approach for picking M150	$OUT[41]	Pick request M150
$IN[42]	Pick completed M150	$OUT[42]	Ready to pick M150

$IN[43]	Authorize to place M180	$OUT[43]	Place request M180
$IN[44]	Clear to go back M180	$OUT[44]	Place completed M180
$IN[45]	Authorize to approach for picking M180	$OUT[45]	Pick request M180
$IN[46]	Pick completed M180	$OUT[46]	Ready to pick M180
$IN[47]	Authorize to place M150	$OUT[47]	Place request M150
$IN[48]	Clear to go back M150	$OUT[48]	Place completed M150
$IN[49]	vision check comp AA	$OUT[49]	ready for vision check AA
$IN[50]	vision check comp AB	$OUT[50]	ready for vision check AB
$IN[51]	vision check comp BB	$OUT[51]	ready for vision check BB
$IN[52]	vision check comp BA	$OUT[52]	ready for vision check BA
$IN[53]		$OUT[53]	
$IN[54]		$OUT[54]	
$IN[55]	NOK Glass + counted	$OUT[55]	NOK Glass picked
$IN[56]	NOK Glass - counted	$OUT[56]	NOK Glass placed
$IN[57]	release plug AA	$OUT[57]	comp spacer plug AA
$IN[58]	release plug AB	$OUT[58]	comp spacer plug AB
$IN[59]	release plug BB	$OUT[59]	comp spacer plug BB
$IN[60]	release m170 BA	$OUT[60]	comp spacer plug BA
$IN[61]	ready for plug AA	$OUT[61]	
$IN[62]	ready for plug AB	$OUT[62]	
$IN[63]	ready for plug BB	$OUT[63]	
$IN[64]	ready for plug BA	$OUT[64]	
$IN[65]	comp check m170 AA1	$OUT[65]	ready for spacer check AA1
$IN[66]	comp check m170 AA2	$OUT[66]	ready for spacer check AA2
$IN[67]	comp check m170 AB	$OUT[67]	ready for spacer check AB
$IN[68]	comp check m170 BB1	$OUT[68]	ready for spacer check BB1
$IN[69]	comp check m170 BB2	$OUT[69]	ready for spacer check BB2
$IN[70]	comp check m170 BA	$OUT[70]	ready for spacer check BA
$IN[71]		$OUT[71]	
$IN[72]		$OUT[72]	
$IN[73]	comp unplug AA	$OUT[73]	at pos spacer unplug AA
$IN[74]	comp unplug AB	$OUT[74]	at pos spacer unplug AB
$IN[75]	comp unplug BB	$OUT[75]	at pos spacer unplug BB
$IN[76]	comp unplug BA	$OUT[76]	at pos spacer unplug BA
$IN[77]	ready for unplug AA	$OUT[77]	
$IN[78]	ready for unplug AB	$OUT[78]	
$IN[79]	ready for unplug BB	$OUT[79]	
$IN[80]	ready for unplug BA	$OUT[80]	
$IN[81]	comp unplug check AA1	$OUT[81]	ready for no spcr chk AA1
$IN[82]	comp unplug check AA2	$OUT[82]	ready for no spcr chk AA1
$IN[83]	comp unplug check AB	$OUT[83]	ready for no spcr chk AB
$IN[84]	comp unplug check BB1	$OUT[84]	ready for no spcr chk BB1
$IN[85]	comp unplug check BB2	$OUT[85]	ready for no spcr chk BB2
$IN[86]	comp unplug check BA	$OUT[86]	ready for no spcr chk BA
$IN[87]	m170 m190 no wait	$OUT[87]	

$IN[88]	no vacuum ok ack	$OUT[88]	no vacuum ok
$IN[89]	spacer check left bit 0	$OUT[89]	a1 act negative bit 0
$IN[90]	spacer check left bit 1	$OUT[90]	a1 act negative bit 1
$IN[91]	spacer check left bit 2	$OUT[91]	a1 act negative bit 2
$IN[92]	spacer check left bit 3	$OUT[92]	a1 act negative bit 3
$IN[93]	spacer check left bit 4	$OUT[93]	a1 act negative bit 4
$IN[94]	spacer check left bit 5	$OUT[94]	a1 act negative bit 5
$IN[95]	spacer check left bit 6	$OUT[95]	a1 act negative bit 6
$IN[96]	spacer check left bit 7	$OUT[96]	a1 act negative bit 7
$IN[97]	spacer check right bit 0	$OUT[97]	a1 act positive bit 0
$IN[98]	spacer check right bit 1	$OUT[98]	a1 act positive bit 1
$IN[99]	spacer check right bit 2	$OUT[99]	a1 act positive bit 2
$IN[100]	spacer check right bit 3	$OUT[100]	a1 act positive bit 3
$IN[101]	spacer check right bit 4	$OUT[101]	a1 act positive bit 4
$IN[102]	spacer check right bit 5	$OUT[102]	a1 act positive bit 5
$IN[103]	spacer check right bit 6	$OUT[103]	a1 act positive bit 6
$IN[104]	spacer check right bit 7	$OUT[104]	a1 act positive bit 7
$IN[105]	CGC Crater Glass Count bit 0	$OUT[105]	vacuum 1 command to plc
$IN[106]	CGC Crater Glass Count bit 1	$OUT[106]	vacuum 2 command to plc
$IN[107]	CGC Crater Glass Count bit 2	$OUT[107]	vacuum 3 command to plc
$IN[108]	CGC Crater Glass Count bit 3	$OUT[108]	vacuum 4 command to plc
$IN[109]	CGC Crater Glass Count bit 4	$OUT[109]	vacuum 5 command to plc
$IN[110]	CGC Crater Glass Count bit 5	$OUT[110]	vacuum 6 command to plc
$IN[111]	CGC Crater Glass Count bit 6	$OUT[111]	vacuum 7 command to plc
$IN[112]	CGC Crater Glass Count bit 7	$OUT[112]	vacuum 8 command to plc
$IN[113]	SGC Scrap Glass Count bit 0	$OUT[113]	vacuum 1 OK to PLC
$IN[114]	SGC Scrap Glass Count bit 1	$OUT[114]	vacuum 2 OK to PLC
$IN[115]	SGC Scrap Glass Count bit 2	$OUT[115]	vacuum 3 OK to PLC
$IN[116]	SGC Scrap Glass Count bit 3	$OUT[116]	vacuum 4 OK to PLC
$IN[117]	SGC Scrap Glass Count bit 4	$OUT[117]	vacuum 5 OK to PLC
$IN[118]	SGC Scrap Glass Count bit 5	$OUT[118]	vacuum 6 OK to PLC
$IN[119]	SGC Scrap Glass Count bit 6	$OUT[119]	vacuum 7 OK to PLC
$IN[120]	SGC Scrap Glass Count bit 7	$OUT[120]	vacuum 8 OK to PLC
$IN[121]	spacer check left 2 bit 0	$OUT[121]	measurement difference X bit 0
$IN[122]	spacer check left 2 bit 1	$OUT[122]	measurement difference X bit 1
$IN[123]	spacer check left 2 bit 2	$OUT[123]	measurement difference X bit 2
$IN[124]	spacer check left 2 bit 3	$OUT[124]	measurement difference X bit 3
$IN[125]	spacer check left 2 bit 4	$OUT[125]	measurement difference X bit 4
$IN[126]	spacer check left 2 bit 5	$OUT[126]	measurement difference X bit 5
$IN[127]	spacer check left 2 bit 6	$OUT[127]	measurement difference X bit 6
$IN[128]	spacer check left 2 bit 7	$OUT[128]	measurement difference X bit sign
$IN[129]	spacer check right 2 bit 0	$OUT[129]	measurement difference Y bit 0
$IN[130]	spacer check right 2 bit 1	$OUT[130]	measurement difference Y bit 1
$IN[131]	spacer check right 2 bit 2	$OUT[131]	measurement difference Y bit 2
$IN[132]	spacer check right 2 bit 3	$OUT[132]	measurement difference Y bit 3

$IN[133]	spacer check right 2 bit 4	$OUT[133]	measurement difference Y bit 4
$IN[134]	spacer check right 2 bit 5	$OUT[134]	measurement difference Y bit 5
$IN[135]	spacer check right 2 bit 6	$OUT[135]	measurement difference Y bit 6
$IN[136]	spacer check right 2 bit 7	$OUT[136]	measurement difference Y bit sign
$IN[137]	offset X bit 0	$OUT[137]	measurement difference Z bit 0
$IN[138]	offset X bit 1	$OUT[138]	measurement difference Z bit 1
$IN[139]	offset X bit 2	$OUT[139]	measurement difference Z bit 2
$IN[140]	offset X bit 3	$OUT[140]	measurement difference Z bit 3
$IN[141]	offset X bit 4	$OUT[141]	measurement difference Z bit 4
$IN[142]	offset X bit 5	$OUT[142]	measurement difference Z bit 5
$IN[143]	offset X bit 6	$OUT[143]	measurement difference Z bit 6
$IN[144]	offset X bit sign	$OUT[144]	measurement difference Z bit sign
$IN[145]	offset Y bit 0	$OUT[145]	
$IN[146]	offset Y bit 1	$OUT[146]	
$IN[147]	offset Y bit 2	$OUT[147]	
$IN[148]	offset Y bit 3	$OUT[148]	
$IN[149]	offset Y bit 4	$OUT[149]	
$IN[150]	offset Y bit 5	$OUT[150]	
$IN[151]	offset Y bit 6	$OUT[151]	
$IN[152]	offset Y bit sign	$OUT[152]	
$IN[153]	offset Z bit 0	$OUT[153]	
$IN[154]	offset Z bit 1	$OUT[154]	
$IN[155]	offset Z bit 2	$OUT[155]	
$IN[156]	offset Z bit 3	$OUT[156]	
$IN[157]	offset Z bit 4	$OUT[157]	
$IN[158]	offset Z bit 5	$OUT[158]	
$IN[159]	offset Z bit 6	$OUT[159]	
$IN[160]	offset Z bit sign	$OUT[160]	
$IN[161]	spacer feedback cylinder 1 bit 0	$OUT[161]	
$IN[162]	spacer feedback cylinder 1 bit 1	$OUT[162]	
$IN[163]	spacer feedback cylinder 1 bit 2	$OUT[163]	
$IN[164]	spacer feedback cylinder 1 bit 3	$OUT[164]	
$IN[165]	spacer feedback cylinder 1 bit 4	$OUT[165]	
$IN[166]	spacer feedback cylinder 1 bit 5	$OUT[166]	
$IN[167]	spacer feedback cylinder 1 bit 6	$OUT[167]	
$IN[168]	spacer feedback cylinder 1 bit 7	$OUT[168]	
$IN[169]	spacer feedback cylinder 2 bit 0	$OUT[169]	
$IN[170]	spacer feedback cylinder 2 bit 1	$OUT[170]	
$IN[171]	spacer feedback cylinder 2 bit 2	$OUT[171]	
$IN[172]	spacer feedback cylinder 2 bit 3	$OUT[172]	
$IN[173]	spacer feedback cylinder 2 bit 4	$OUT[173]	
$IN[174]	spacer feedback cylinder 2 bit 5	$OUT[174]	
$IN[175]	spacer feedback cylinder 2 bit 6	$OUT[175]	
$IN[176]	spacer feedback cylinder 2 bit 7	$OUT[176]	

$IN[177]	spacer feedback cylinder 3 bit 0	$OUT[177]	
$IN[178]	spacer feedback cylinder 3 bit 1	$OUT[178]	
$IN[179]	spacer feedback cylinder 3 bit 2	$OUT[179]	
$IN[180]	spacer feedback cylinder 3 bit 3	$OUT[180]	
$IN[181]	spacer feedback cylinder 3 bit 4	$OUT[181]	
$IN[182]	spacer feedback cylinder 3 bit 5	$OUT[182]	
$IN[183]	spacer feedback cylinder 3 bit 6	$OUT[183]	
$IN[184]	spacer feedback cylinder 3 bit 7	$OUT[184]	
$IN[185]	spacer feedback cylinder 4 bit 0	$OUT[185]	
$IN[186]	spacer feedback cylinder 4 bit 1	$OUT[186]	
$IN[187]	spacer feedback cylinder 4 bit 2	$OUT[187]	
$IN[188]	spacer feedback cylinder 4 bit 3	$OUT[188]	
$IN[189]	spacer feedback cylinder 4 bit 4	$OUT[189]	
$IN[190]	spacer feedback cylinder 4 bit 5		
$IN[191]	spacer feedback cylinder 4 bit 6	$OUT[300]	BLOW_ALL
$IN[192]	spacer feedback cylinder 4 bit 7	$OUT[301]	VACUUM_ALL
…		…	
$IN[1025]	Predefined input always in HIGH state.	$OUT[1025]	modul_1
$IN[1026]	Predefined input always in LOW state.	$OUT[1026]	
$IN[1027]		$OUT[1027]	modul_2
$IN[1028]		$OUT[1028]	
$IN[1029]		$OUT[1029]	modul_3
$IN[1030]		$OUT[1030]	
$IN[1031]		$OUT[1031]	
$IN[1032]		$OUT[1032]	
$IN[1033]		$OUT[1033]	
$IN[1034]		$OUT[1034]	
$IN[1035]		$OUT[1035]	
$IN[1036]		$OUT[1036]	
$IN[1037]		$OUT[1037]	
$IN[1038]		$OUT[1038]	
$IN[1039]		$OUT[1039]	
$IN[1040]	vacuum_1	$OUT[1040]	
$IN[1041]	vacuum_3	$OUT[1041]	
$IN[1042]	sens2	$OUT[1042]	
$IN[1043]		$OUT[1043]	vacuum_1
$IN[1044]		$OUT[1044]	vacuum_2
$IN[1045]		$OUT[1045]	vacuum_3
$IN[1046]		$OUT[1046]	vacuum_4
$IN[1047]		$OUT[1047]	vacuum_5
$IN[1048]	vacuum_2	$OUT[1048]	
$IN[1049]	vacuum_4	$OUT[1049]	
$IN[1050]	sens4	$OUT[1050]	
$IN[1051]		$OUT[1051]	blow_1

$IN[1052]		$OUT[1052]	blow_2
$IN[1053]		$OUT[1053]	blow_3
$IN[1054]		$OUT[1054]	blow_4
$IN[1055]		$OUT[1055]	blow_5
$IN[1056]	vacuum_5	$OUT[1056]	
$IN[1057]	vacuum_7	$OUT[1057]	
$IN[1058]	sens1	$OUT[1058]	
$IN[1059]		$OUT[1059]	
$IN[1060]		$OUT[1060]	vacuum_6
$IN[1061]		$OUT[1061]	vacuum_7
$IN[1062]		$OUT[1062]	vacuum_8
$IN[1063]		$OUT[1063]	blow_enable
$IN[1064]	vacuum_6	$OUT[1064]	
$IN[1065]	vacuum_8	$OUT[1065]	
$IN[1066]	sens3	$OUT[1066]	
$IN[1067]		$OUT[1067]	
$IN[1068]		$OUT[1068]	blow_6
$IN[1069]		$OUT[1069]	blow_7
$IN[1070]		$OUT[1070]	blow_8
$IN[1071]		$OUT[1071]	fix_cups
$IN[1072]	AI01_MSB bit 0	$OUT[1072]	analog port 0 reset
$IN[1073]	AI01_MSB bit 1	$OUT[1073]	analog port 1 reset
$IN[1074]	AI01_MSB bit 2	$OUT[1074]	analog port 3 reset
$IN[1075]	AI01_MSB bit 3	$OUT[1075]	analog port 4 reset
$IN[1076]	AI01_MSB bit 4		
$IN[1077]	AI01_MSB bit 5		
$IN[1078]	AI01_MSB bit 6		
$IN[1079]	AI01_MSB bit 7		
$IN[1080]	AI01_LSB bit 0		
$IN[1081]	AI01_LSB bit 1		
$IN[1082]	AI01_LSB bit 2		
$IN[1083]	AI01_LSB bit 3		
$IN[1084]	AI01_LSB bit 4		
$IN[1085]	AI01_LSB bit 5		
$IN[1086]	AI01_LSB bit 6		
$IN[1087]	AI01_LSB bit 7		
$IN[1088]	AI02_MSB bit 0		
$IN[1089]	AI02_MSB bit 1		
$IN[1090]	AI02_MSB bit 2		
$IN[1091]	AI02_MSB bit 3		
$IN[1092]	AI02_MSB bit 4		
$IN[1093]	AI02_MSB bit 5		
$IN[1094]	AI02_MSB bit 6		
$IN[1095]	AI02_MSB bit 7		
$IN[1096]	AI02_LSB bit 0		
$IN[1097]	AI02_LSB bit 1		

$IN[1098]	AI02_LSB bit 2		
$IN[1099]	AI02_LSB bit 3		
$IN[1100]	AI02_LSB bit 4		
$IN[1101]	AI02_LSB bit 5		
$IN[1102]	AI02_LSB bit 6		
$IN[1103]	AI02_LSB bit 7		
$IN[1104]	AI03_MSB bit 0		
$IN[1105]	AI03_MSB bit 1		
$IN[1106]	AI03_MSB bit 2		
$IN[1107]	AI03_MSB bit 3		
$IN[1108]	AI03_MSB bit 4		
$IN[1109]	AI03_MSB bit 5		
$IN[1110]	AI03_MSB bit 6		
$IN[1111]	AI03_MSB bit 7		
$IN[1112]	AI03_LSB bit 0		
$IN[1113]	AI03_LSB bit 1		
$IN[1114]	AI03_LSB bit 2		
$IN[1115]	AI03_LSB bit 3		
$IN[1116]	AI03_LSB bit 4		
$IN[1117]	AI03_LSB bit 5		
$IN[1118]	AI03_LSB bit 6		
$IN[1119]	AI03_LSB bit 7		
$IN[1120]	AI04_MSB bit 0		
$IN[1121]	AI04_MSB bit 1		
$IN[1122]	AI04_MSB bit 2		
$IN[1123]	AI04_MSB bit 3		
$IN[1124]	AI04_MSB bit 4		
$IN[1125]	AI04_MSB bit 5		
$IN[1126]	AI04_MSB bit 6		
$IN[1127]	AI04_MSB bit 7		
$IN[1128]	AI04_LSB bit 0		
$IN[1129]	AI04_LSB bit 1		
$IN[1130]	AI04_LSB bit 2		
$IN[1131]	AI04_LSB bit 3		
$IN[1132]	AI04_LSB bit 4		
$IN[1133]	AI04_LSB bit 5		
$IN[1134]	AI04_LSB bit 6		
$IN[1135]	AI04_LSB bit 7		
…	…		
$IN[1168]	Pin 1 SC Wire break Port 0		
$IN[1169]	Pin 1 SC Wire break Port 1		
$IN[1170]	Pin 1 SC Wire break Port 2		
$IN[1171]	Pin 1 SC Wire break Port 3		
$IN[1172]	Pin 1 SC Wire break Port 4		
$IN[1173]	Pin 1 SC Wire break Port 5		
$IN[1174]	Pin 1 SC Wire break Port 6		

$IN[1175]	Pin 1 SC Wire break Port 7		
$IN[1176]	Underflow Port 0		
$IN[1177]	Underflow Port 1		
$IN[1178]	Underflow Port 2		
$IN[1179]	Underflow Port 3		
$IN[1180]	Underflow Port 4		
$IN[1181]	Underflow Port 5		
$IN[1182]	Underflow Port 6		
$IN[1183]	Underflow Port 7		
$IN[1184]	Overflow Port 0		
$IN[1185]	Overflow Port 1		
$IN[1186]	Overflow Port 2		
$IN[1187]	Overflow Port 3		
$IN[1188]	Overflow Port 4		
$IN[1189]	Overflow Port 5		
$IN[1190]	Overflow Port 6		
$IN[1191]	Overflow Port 7		
$IN[1192]	Undervoltage Us		
$IN[1193]			
$IN[1194]			
$IN[1195]			
$IN[1196]			
$IN[1197]			
$IN[1198]			
$IN[1199]			
$IN[1200]			

Figure 9 List of inputs and outputs

2.4. Robot sensor setup

Optic Laser sensors setup parameters.

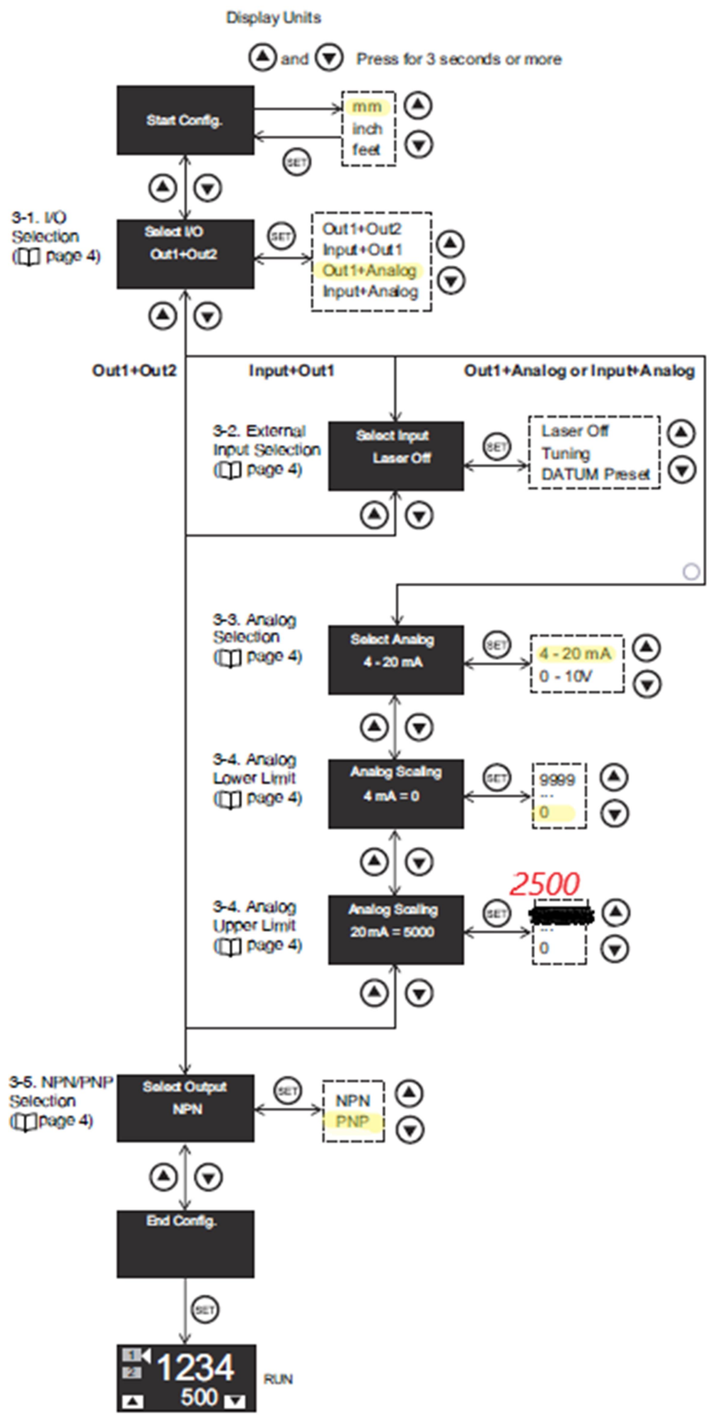
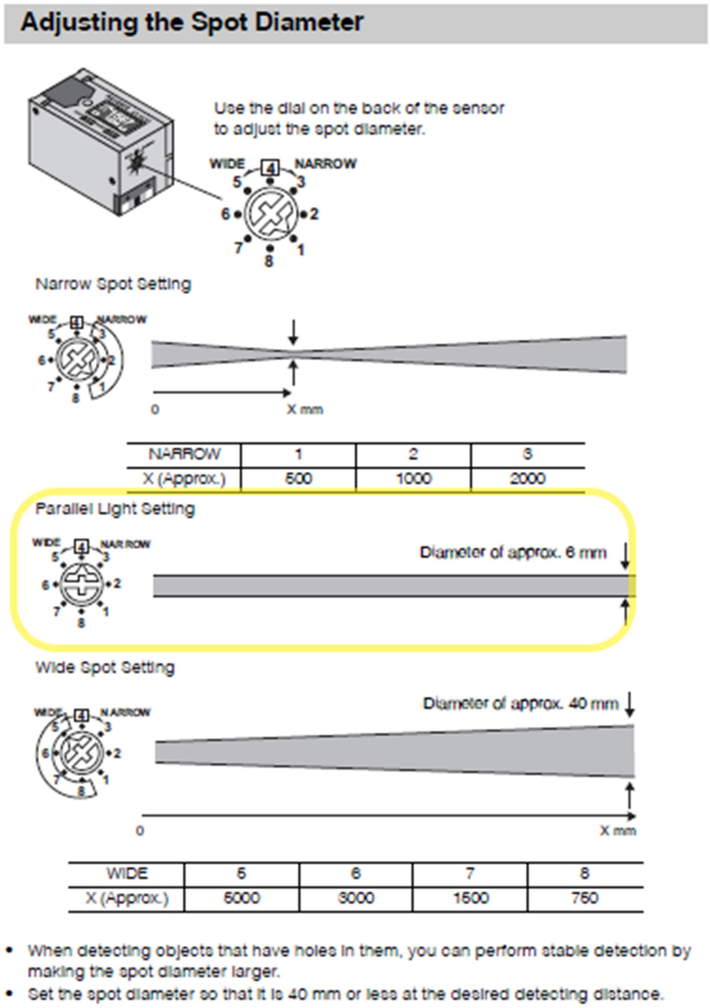


Figure 10 Optic laser setup

2.5. Robot sensor calibration

Optic laser light width (narrow to wide)


Adjusting the Spot Diameter

Use the dial on the back of the sensor to adjust the spot diameter.

Narrow Spot Setting

NARROW	1	2	3
X (Approx.)	500	1000	2000

Parallel Light Setting

Wide Spot Setting

WIDE	5	6	7	8
X (Approx.)	5000	3000	1500	750

- When detecting objects that have holes in them, you can perform stable detection by making the spot diameter larger.
- Set the spot diameter so that it is 40 mm or less at the desired detecting distance.

Figure 11 Optic laser sensor diameter calibration

■ **LR-TB5000CL**

Detecting distance [mm]	Repetition Accuracy [mm] (Typical) (Under stable temperature)				
	White Paper (Reflectivity: 90%)				
	Response Time[ms]				
	2	20	50	200	2000
1000	±9	±6	±6	±3	±3
5000	±42	±14	±10	±5	±5

Detecting distance [mm]	Repetition Accuracy [mm] (Typical) (Under stable temperature)				
	Gray Paper (Reflectivity: 18%)				
	Response Time[ms]				
	2	20	50	200	2000
1000	±12	±6	±4	±3	±3
5000	±159	±42	±31	±15	±8

Figure 12 Optic laser sensor response time setup

Ultrasonic far distance calibration.

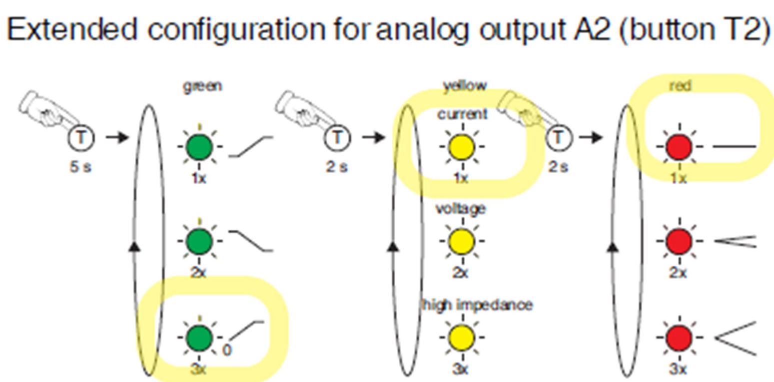


Figure 13 Ultrasonic sensor setup parameters

Zero Point Line is used for far distance calibration. Any value set for SP1 is ignored. To teach SP2 “calib.src” robot program must be used. When robot reaches its programmed position press and hold T2 button briefly while putting an object close to sensor (SP1 will be set, since the value for SP1 will be ignored distance of the object does not matter.) Remove any object between the wall and the sensor. Press and hold T2 button (approx. 2s). SP2 will be set. If green LED flashes 3x times it indicates setting is completed, if red LED flashes 3x times it indicates fault occurred, repeat setting. The value that ultrasonic sensor sends at this position must be equal to 1453.

3. Robot Restore Backup

As mention on Kuka KRC4 user manual.

4. Software Structure

4.1. Main Program

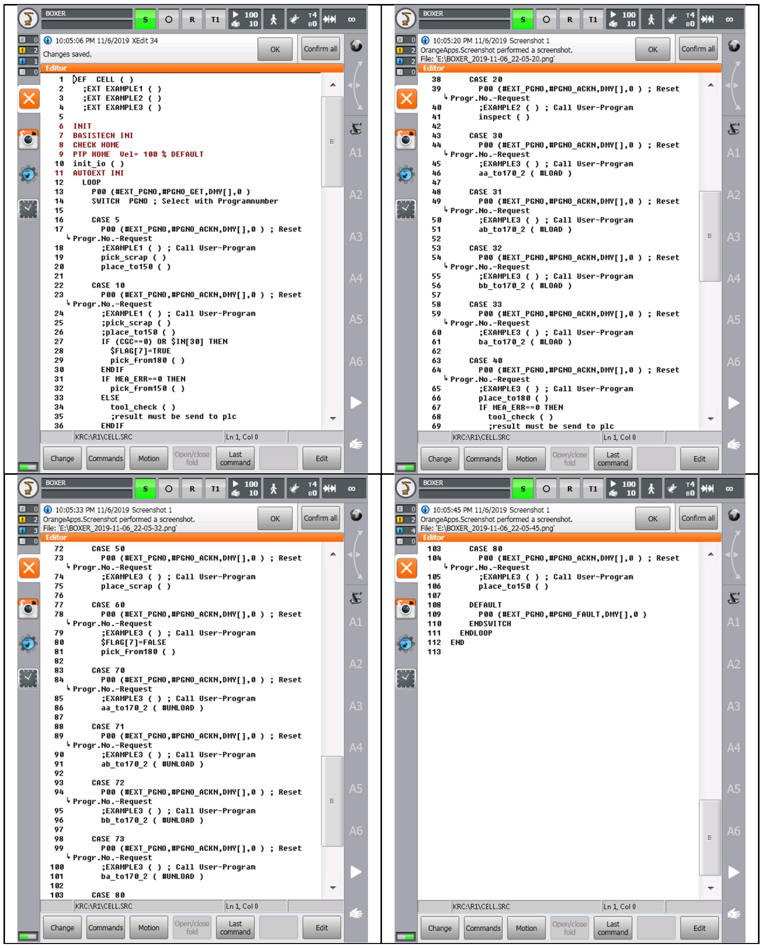


Figure 14 Cell.src

Cell.src is using a main control program. It waits for a program number from PLC then execute in continuous loop. Each program number execute one or two individual different operations. General sequence of loading and unloading case control by PLC

4.2. SPS Program

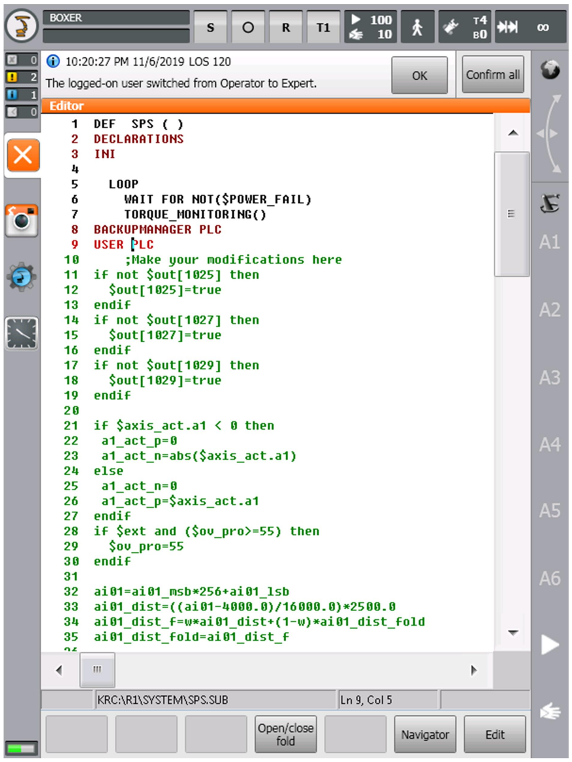


Figure 15 Summit interpreter

Robot summit interpreter is running in continuous mode. Getting analog values, sending robot A1 axis positon to PLC, Speed limit for auto-external mode, sending power for I/O link master model channels, Quick blow and vacuum function with $out[300] and $out[301].

4.3. Paths

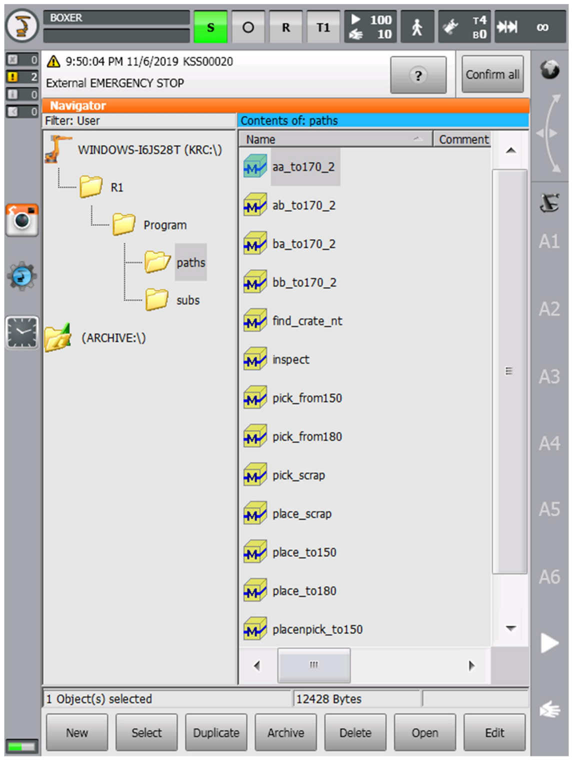


Figure 16 Paths

4.3.1. aa_to170_2

Plug and unplug operation at the long bottom edge of MB while robot at home position. This program divides into 2 operations. Program will execute these 2 operations depending on loading and unloading mode. For loading mode robot will wait signal $IN 87 to move plug position. After reaching plug position robot will send signal $OUT 57. This signal will trigger PLC to start plugging process. After PLC finishes plugging, PLC will send signal $IN 87 to send robot to sensor position to check if plug operation is successfully done. After reaching sensor position robot will send $OUT 65 to PLC. This signal will trigger PLC to start reading sensor values. After PLC finishes reading sensor values, PLC will send signal $IN 87 to robot. Robot will leave sensor position. If PLC sends signal $IN 77 and doesn't send signal $IN87(which happens when plug operation is unsuccessful according to sensor values) Robot will try to unplug spacers and start plugging operation again. Robot will move to home position if sensor check is successful. For unloading mode robot will wait signal $IN 87 or $IN 77 to move check unplug position. After moving check unplug position robot will move 5mm in z axis towards M170 and check if average of lift cylinder sensor value is less than 94. If after 6 trials average of lift cylinder sensor value is not less than 94 robot sends signal $OUT 34 to PLC. If average of lift cylinder sensor value is less than 94 robot will send signal $OUT 73 and calculate x_offset. Calculation of x_offset is [("average of lift cylinder sensor values"-97.0)*0.833]+("trial number"*0.5). $OUT34 will trigger PLC to start unplugging process. After PLC finishes unplugging, PLC will send signal $IN 87 to send robot to sensor position to check if unplug operation is successfully done. After reaching sensor position with offset obtained from calculation robot will send $OUT 65 to PLC. This signal will trigger PLC to start reading sensor values.

After PLC finishes reading sensor values, PLC will send signal $IN 87 to robot. Robot will leave sensor position. If PLC sends signal $IN 77 and doesn't send signal $IN87(which happens when unplug operation is unsuccessful according to sensor values) Robot will start unplug operation again. Robot will move to home position if sensor check is successful.

4.3.2. ab_to170_2

Plug and unplug operation at the short east edge of MB while robot at home position. This program divides into 2 operations. Program will execute these 2 operations depending on loading and unloading mode. For loading mode robot will wait signal $IN 87 to move plug position. After reaching plug position robot will send signal $OUT 58. This signal will trigger PLC to start plugging process. After PLC finishes plugging, PLC will send signal $IN 87 to send robot to sensor position to check if plug operation is successfully done. After reaching sensor position robot will send $OUT 67 to PLC. This signal will trigger PLC to start reading sensor values. After PLC finishes reading sensor values, PLC will send signal $IN 87 to robot. Robot will leave sensor position. If PLC sends signal $IN 78 and doesn't send signal $IN87(which happens when plug operation is unsuccessful according to sensor values) Robot will try to unplug spacers and start plugging operation again. Robot will move to home position if sensor check is successful . For unloading mode robot will wait signal $IN 87 or $IN 78 to move check unplug position.After moving check unplug position robot will move 5mm in z axis towards M170 and check if average of lift cylinder sensor value is less than 94. If after 6 trials average of lift cylinder sensor value is not less than 94 robot sends signal $OUT 74 to PLC. If average of lift cylinder sensor value is less than 94 robot will send signal $OUT 73 and calculate x_offset. Calculation of x_offset is [("average of lift cylinder sensor values"-97.0)*0.833]+("trial number"*0.5). $OUT 74 will trigger PLC to start unplugging process. After PLC finishes unplugging, PLC will send signal $IN 87 to send robot to sensor position to check if unplug operation is successfully done. After reaching sensor position with offset obtained from calculation robot will send $OUT 65 to PLC. This signal will trigger PLC to start reading sensor values. After PLC finishes reading sensor values, PLC will send signal $IN 87 to robot. Robot will leave sensor position. If PLC sends signal $IN 78 and doesn't send signal $IN87(which happens when unplug operation is unsuccessful according to sensor values) Robot will start unplug operation again. Robot will move to home position if sensor check is successful.

4.3.3. ba_to170_2

Plug and unplug operation at the short west edge of MB while robot at home position. This program divides into 2 operations. Program will execute these 2 operations depending on loading and unloading mode. For loading mode robot will wait signal $IN 87 to move plug position. After reaching plug position robot will send signal $OUT 60. This signal will trigger PLC to start plugging process. After PLC finishes plugging, PLC will send signal $IN 87 to send robot to sensor position to check if plug operation is successfully done. After reaching sensor position robot will send $OUT 70 to PLC. This signal will trigger PLC to start reading sensor values. After PLC finishes reading sensor values, PLC will send signal $IN 87 to robot. Robot will leave sensor position. If PLC sends signal $IN 80 and doesn't send signal $IN87(which happens when plug operation is unsuccessful according to sensor values) Robot will try to unplug spacers and start plugging operation again. Robot will move to home position if sensor check is successful . For unloading mode robot will wait signal $IN 87 or $IN 80 to move check unplug position. After moving check unplug position robot will move 5mm in z axis towards M170 and check if average of lift cylinder sensor value is less than 94. If after 6 trials average of lift cylinder sensor value is not less than 94 robot sends signal $OUT 34 to PLC. If average of lift cylinder sensor value is less than 94 robot will send signal $OUT 77 and calculate x_offset. Calculation of x_offset is [("average of lift cylinder sensor values"-97.0)*0.833]+("trial number"*0.5). $OUT 77 will trigger PLC to start unplugging process.. After PLC finishes unplugging, PLC will send signal $IN 87 to send robot to sensor position to check if unplug operation is successfully done. After reaching sensor position with offset obtained from calculation robot will send $OUT 70 to PLC. This signal will trigger PLC to start reading sensor values. After PLC finishes reading sensor values, PLC will send signal $IN 87 to robot. Robot will leave sensor position. If PLC sends signal $IN 80 and doesn't send signal $IN87(which happens when unplug operation is unsuccessful according to sensor values) Robot will start unplug operation again. Robot will move to home position if sensor check is successful.

4.3.4. bb_to170_2

Plug and unplug operation at the long top edge of MB while robot at home position. This program divides into 2 operations. Program will execute these 2 operations depending on loading and unloading mode. For loading mode robot will wait signal $IN 87 to move plug position. After reaching plug position robot will send signal $OUT 59. This signal will trigger PLC to start plugging process. After PLC finishes plugging, PLC will send signal $IN 87 to send robot to sensor position to check if plug operation is successfully done. After reaching sensor position robot will send $OUT 68 to PLC. This signal will trigger PLC to start reading sensor values. After PLC finishes reading sensor values, PLC will send signal $IN 87 to robot. Robot will leave sensor position. If PLC sends signal $IN 79 and doesn't send signal $IN87(which happens when plug operation is unsuccessful according to sensor values) Robot will try to unplug spacers and start plugging operation again. Robot will move to home position if sensor check is successful . For unloading mode robot will wait signal $IN 87 or $IN 79 to move check unplug position. After moving check unplug position robot will move 5mm in z axis towards M170 and check if average of lift cylinder sensor value is less than 94. If after 6 trials average of lift cylinder sensor value is not less than 94 robot sends signal $OUT 34 to PLC. If average of lift cylinder sensor value is less than 94 robot will send signal $OUT 75 and calculate x_offset. Calculation of x_offset is [("average of lift cylinder sensor values"-97.0)*0.833]+("trial number"*0.5). $OUT 75 will trigger PLC to start unplugging process. After PLC finishes unplugging, PLC will send signal $IN 87 to send robot to sensor position to check if unplug operation is successfully done. After reaching sensor position with offset obtained from calculation robot will send $OUT 68 to PLC. This signal will trigger PLC to start reading sensor values. After PLC finishes reading sensor values, PLC will send signal $IN 87 to robot. Robot will leave sensor position. If PLC sends signal $IN 79 and doesn't send signal $IN87(which happens when unplug operation is unsuccessful according to sensor values) Robot will start unplug operation again. Robot will move to home position if sensor check is successful .

4.3.5. find_crate_nt

Searching and measurement of crate and calculate the offset vector for picking and placing operations of turn table. There are 3 work types for this program. Load, Unload and Check are the work types. Find_crate_nt is called by other path programs to calculate position of the MB on the crate or position of the crate on the twister M180. Also find_crate_nt is used for verifying MB count on the crate with calculations on the robot. Find_crate_nt should always be called from other path programs to be able to measure and calculate correct values for picking and placing glass. The program starts with $IN 39 which indicates Side A or Side B of twister on robot position. If $IN 39 is true side is B and if $IN 39 is false side is A. For A side value of SIDE=0 and for B side value of SIDE=1 For A side XP10 position is used and for side B side XP6 position is used. If PLC sends signal $IN 30(The crate is new, therefore there was no measurement done for this crate before) and work type is UNLOAD or MB count on twister is 0 and work type is CHECK robot starts to move on Y axis. Interrupt 7 is used for finding edge of MB on Y axis. For empty crate if the difference between ai01_dist_old(value of distance sensor 1 at starting position) and ai01_dist is bigger than 25 FLAG[1] sets to false and found_y1 ( ) is called, for crate with MB on it if the difference between ai01_dist_old(value of distance sensor 1 at starting position) and ai01_dist is bigger than 0 FLAG[1] sets to false and found_y1 ( ) is called . Found_y1 stops the robot and writes current Y position of robot to YY1[3+SIDE].XY. If the difference between ai01_dist_old(value of distance sensor 1 at starting position) and ai01_dist is not bigger than 25 or 10(depending on if crate is empty or not) after robot moves 100mm on Y axis FLAG[1] remains true. After robot finishes its movement on Y axis, robot starts to move on X axis. Interrupt 8 is used for finding edge of MB on X axis. For empty crate the difference between ai02_dist_old(value of distance sensor 2 at starting position) and ai02_dist(current value of distance sensor 2) is bigger than 25 FLAG[2] sets to false and found_x1 ( ) is called, for crate with MB on it if the difference between ai02_dist_old(value of distance sensor 2 at starting position) and ai02_dist(current value of distance sensor 2) is bigger than 0 FLAG[2] sets to false and found_x1 ( ) is called. Found_x1 stops the robot and writes current X position of robot to XX1[3+SIDE].XY. If the difference between ai02_dist_old(value of distance sensor 2 at starting position) and ai02_dist is not bigger than 25 or 10(depending on if crate is empty or not) after robot moves 100mm on X axis FLAG[2] remains true. Robot moves -20mm on X axis and calculates offsets. If the current glass count on twister is 0 and ai04_dist_f(current value of the ultrasonic sensor) is bigger than DEEP[SIDE+1,1](constant value of deep of the crate when 1 glass was placed on a perfectly placed crate that was measured with ultrasonic sensor) robot writes current values of sensor 1 and sensor 2 to YY1[3+SIDE] and XX1[3+SIDE] in respective order. The difference between YY1[3+SIDE] and YY1[1+SIDE](a constant value of Y position of crate that was measured with sensors on a perfectly placed crate on twister M180) is value of the offset on Y axis. The difference between XX1[3+SIDE] and XX1[1+SIDE](a constant value of X position of crate that was measured with sensors on a perfectly placed crate on twister M180) is value of the offset on X axis and robot sends signal $OUT 30(measurement done) to plc. If ai04_dist_f(current value of the ultrasonic sensor) is less than DEEP[SIDE+1,1](constant value of deep of the crate when 1 glass was placed on a perfectly placed crate that was measured with ultrasonic sensor) or FLAG[1] or FLAG[2] is still true robot will send signal $OUT 31(measurement error) to PLC and return home position. If the current glass count is different than 0, difference between DEEP[SIDE+1,1](measured value of the deepness of the crate with 1 MB on crate) and DEEP[SIDE+1,CGC](measured value of the deepness of the crate with current MB count on crate) is equal to MEA_DEEP. Z offset is equal to MEA_DEEP-11.90. If difference between ("current MB count"*11.80)-MEA_DEEP is not between +6,-6 robot sends signal $OUT 31(measurement error) to PLC, if not robot sends signal $OUT 3(measurement done) to PLC. If both conditions of PLC sends signal $IN 30(The crate is new, therefore there was no measurement done for this crate before) and work type is UNLOAD or MB count on twister is 0 and work type is CHECK are not satisfied, current value of the ultrasonic sensor is written to DEEP[SIDE+1,CGC](measured value of the deepness of the crate with current MB count on crate) and MEA_DEEP is equal to difference between DEEP[SIDE+1,1](constant value of deep of the crate when 1 glass was placed on a perfectly placed crate that was measured with ultrasonic sensor) and DEEP[SIDE+1,CGC] (measured value of the deepness of the crate with current MB count on crate). If work type is LOAD then Z offset is equal to M180_ZERO_OFFSET[SIDE+1].Z(measured value of empty crate)-("current MB count on crate"*11.95). After calculation robot sends signal $IN 30(measurement done) to PLC. If work type is unload Z offset is equal to (MEA_DEEP-11.95)+CRATE_DEEP_OFF. If difference between ("current MB count"*11.80)-MEA_DEEP is not between

+6,-6 robot sends signal $OUT 31(measurement error) to PLC. Program also sends X, Y, Z offset values to PLC after calculations and it is shown on HMI screen.

4.3.6. Inspect

Vision check for all edge at once. Robot will start with long top edge of MB. Robot will move to camera position and send signal $OUT 49 after reaching camera position. This will trigger PLC to start camera inspection process. PLC will send signal $IN 87 after camera inspection is completed. Robot will move to starting position. Then robot will start with short east edge of MB . Robot will move to camera position and send signal $OUT 50 after . This will trigger PLC to start camera inspection process. PLC will send signal $IN 87 after camera inspection is completed. Robot will move to starting position. Robot will start with long bottom edge of MB. Robot will move to camera position and send signal $OUT 51 after reaching camera position. This will trigger PLC to start camera inspection process. PLC will send signal $IN 87 after camera inspection is completed. Robot will move to starting position. Robot will start with short west edge of MB. Robot will move to camera position and send signal $OUT 52 after reaching camera position. This will trigger PLC to start camera inspection process. PLC will send signal $IN 87 camera inspection is completed. Robot will move to starting position. Robot will move to home position.

4.3.7. pick_from150

picking operation from M150 lifting conveyor. Robot will wait for signal $IN 35 from PLC to start operation. $IN 35 indicates that M150 lifting conveyor is at robot position. Robot will send signal $OUT 41 to PLC to ask for permission to approach M150 lifting conveyor. After PLC sends signal $IN 41 robot will be permitted to approach M150 lifting conveyor. Robot will stop sending signal $OUT 35 to PLC to disable any movement on M150 lifting conveyor. Robot will move to M150 lifting conveyor pick up position. Robot will call vacuum_onwt ( ) sub program. Robot will check if $IN 1065, $IN 1064, $IN 1056, $IN 1049, $IN 1048 and $IN 1041 signals are true. These signals are obtained from vacuum sensor feedback. Then Robot will send signal $OUT 42 to PLC for permission to pick up MB from M150 lifting conveyor if $IN 1065, $IN 1064, $IN 1056, $IN 1049, $IN 1048 and $IN 1041 signals are true . Robot will wait for signal $IN 42 from PLC to pick MB from M150 lifting conveyor. When PLC sends signal $IN42 robot will pick up glass and move to home position. If $IN 1065, $IN 1064, $IN 1056, $IN 1049, $IN 1048 and $IN 1041 signals are false robot will call vacuum_off ( ), blow_on ( ), blow_check ( ), blow_off ( ) sub programs in respective order. Then robot will move to home position.

4.3.8. pick_from180

Picking operation from M180 Turn table side A and also Side B. This program will be called with 2 different work types. If work type is check, program will only be used to check MB position on M180 twister, if work type is unload program will be used for placing MB on M180 twister. Robot will wait signal $IN 39 or $IN 38 from PLC. $IN 38 indicates twister side a and $IN 39 indicates twister side b. After PLC sends $IN 39 or $IN 38 robot will stop sending $OUT 38 to PLC to prevent PLC to move M180 twister. Robot will send $OUT 45 to request for permission to approach twister. Then robot will call find_crate_nt(wt) path program. If current glass count from PLC is different than 0 and there was no measurement error from robot sensors after find_crate_nt(wt) program robot will move to picking glass from twister position(if side is A XP6 point will be used, if side is B XP13 point will be used). vacuum_on_56_wt ( ) sub program will be called. If $IN 1056 and $IN 1064 are true (vacuum sensor feedback) robot will finish vacuum operation. If not robot will move 2mm towards twister and try vacuum again and check signal $IN 1056 and $IN 1064.Robot will try this 2 times. If vacuum sensor feedback signal $IN 1056 or $IN 1064 are false, robot will send signal $OUT 88 to notify PLC that vacuum is not ok. After PLC sends signal $IN 88(PLC acknowledged vacuum is not ok) robot will call vacuum_off( ), blow_on ( ) and move 10mm away from twister and call blow_check ( ) and blow_off ( ) sub programs and move to home position. If signal $IN 1056 and $IN 1064 are true robot will send signal $OUT 46 to PLC to ask permission to pick MB up from twister. After PLC sends signal $IN 46 Robot will move to home position with MB.

4.3.9. pick_scrap

Picking from the scrap rack according to number of MB which is sending by PLC. This program is not called not in auto external mode. Only for manual operations. Robot will multiply MB count on scrap rack with 2.39 and subtract it from 505 to find out z axis offset value. Robot will move towards scrap rack according to z axis offset obtained from calculation. Vacuum_on_wt ( ) sub program will be called. If $IN 1065, $IN 1064, $IN 1056, $IN 1049, $IN 1048 and $IN 1041 signals are true robot will pick the MB and move to home position. If any of $IN 1065, $IN 1064, $IN 1056, $IN 1049, $IN 1048 and $IN 1041 signals are false vacuum_off ( ), blow_off ( ) , blow_check ( ) sub programs will be called respectively. Robot will move home position.

4.3.10. place_scrap

Placing to the scrap rack according to number of MB which is sending by PLC. Robot will multiply MB count on scrap rack with 2.4 and subtract it from 500. Robot will move to place MB to scrap position. Robot will send signal $OUT 56 to PLC to notify robot is on place scrap position. After PLC sends signal $IN 56 robot will call vacuum_off ( ), blow_on ( ), blow_check ( ) sub programs in respective order and move 50mm away from scrap rack and call blow_off ( ) sub program. Robot will move move to home position.

4.3.11. place_to150

Placing MB to the M150 lift conveyor. Robot will wait for signal $IN 35 from PLC which indicates M150 lifting conveyor is at robot position. After PLC sends $IN 35 robot will stop sending $OUT 35 to PLC to prevent PLC moving M150 lifting conveyor. Robot will move to M150 lifting conveyor place position. Vacuum_off ( ), blow_on ( ), blow_check ( ), blow_off ( ) sub programs will be called in respective order. Robot will send signal $OUT 48 to PLC to notify robot placed MB on M150 lifting conveyor. After PLC sends signal $IN 48 (Clear to go back from M180), robot will move to home position.

4.3.12. place_to180

Placing MB to twister (turn table) A and B sides. Robot will wait for signal $IN 39(twister side A) or $IN 38(twister side b) from PLC. After PLC sends signal $IN39 or $IN 38, robot will stop sending signal $OUT 38 to prevent PLC from moving twister. Robot will move to place MB to 180 position and send signal $OUT 44 to PLC to notify robot reached place 180 position. After PLC sends signal $IN 44, robot will call vacuum_off ( ), blow_on ( ), blow_check ( ) sub programs in respective order and move 50mm away from twister in z axis and call blow_off ( ) sub program. Find_crate_nt (#LOAD) path program will be called. Then robot will move to home position.

4.3.13. picknplace_to150

(optional) for future use only if necessary to align to MANZ.

4.4. Sub Routines

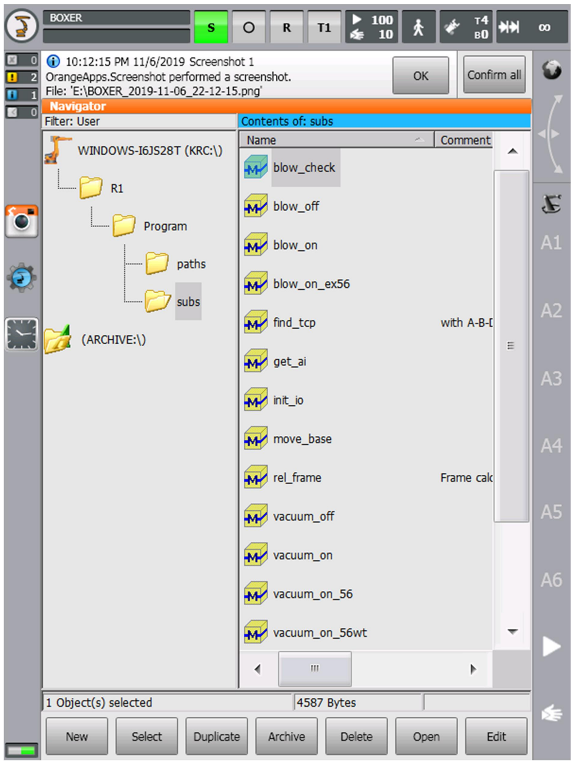


Figure 17 Sub routines

Sub routines for general use; blow and vacuum operations.

4.5. Special Programs

4.5.1. Tool_check

Tool_check; checking sensor positon and also orientation.

4.5.2. Sensor Calibration

Calib; Ultrasonic sensor calibration purpose after changing sensor.

4.5.3. Home

Home; Home to MB lifting positon halt and turn back to home. Manuel use only.

4.5.4. Maintenance

Maintenance; Robot going to maintenance positon and halt. Use manually.

5. Base

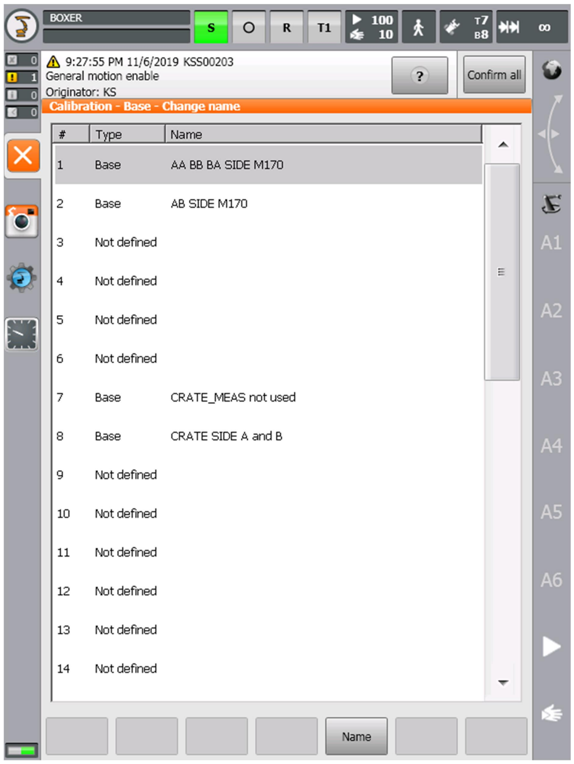


Figure 18 Base

- Base_Data [1] M170 East Side using in aa_to170, bb_to170, ba_to170
- Base_Data [2] M170 West Side using in ab_to170
- Base_Data [8] M180 Chancing Side A and Side B
- Base_Data [6] Scrap Rack

Figure 19 Base Data[8]

Blue = X Axis
Red = Y Axis
Black = Z Axis

6. Tool

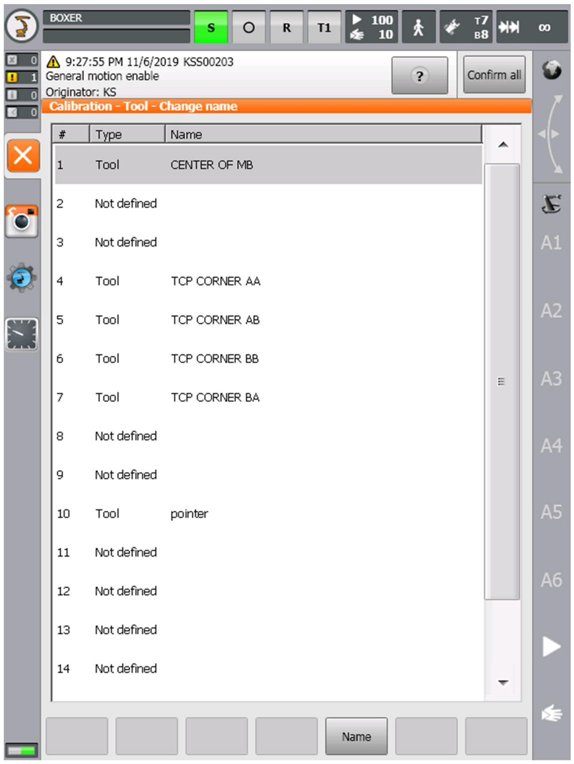


Figure 20 Tool

- Tool_Data [1] General only Z offset
- Tool_Data [4] AA Side East Corner, aa_to170, pick_from180, place_to180, place_toscrap
- Tool_Data [5] ab_to170
- Tool_Data [6] bb_to170
- Tool_Data [7] ba_to170

Figure 21 Tool Data [4] and Base Data[1]

Blue = X Axis
Red = Y Axis
Black = Z Axis

7. References

Figure 1 Robot Mastering Type........3
Figure 2 Ethernet IP configuration........4
Figure 3 Ethernet IP Slave configuration........5
Figure 4 Workvisual I/O Ethernet IP config........6
Figure 5 I/O link master model........7
Figure 6 Analog mode........7
Figure 7 Proses data format........8
Figure 8 Pin assignment........8
Figure 9 List of inputs and outputs........16
Figure 10 Optic laser setup........17
Figure 11 Optic laser sensor diameter calibration........18
Figure 12 Optic laser sensor response time setup........19
Figure 13 Ultrasonic sensor setup parameters........19
Figure 14 Cell.src........20
Figure 15 Summit interpreter........21
Figure 16 Paths........22
Figure 17 Sub routines........28
Figure 18 Base........29
Figure 19 Base Data[8]........30
Figure 20 Tool........31
Figure 21 Tool Data [4] and Base Data[1]........32