## UŽIVATELSKÁ PŘÍRUČKA

1	Kolaborativní robot
	1.1 Mattee produktu
	1.2 Rychie spusteni
	1.2. Instalace a zapnuti robota
	1.2.1.1 Install the robot arm
	1.2.1.2Připojení ovládací skříňky
	1.2.1.       3Znát pole tlačítek a koncovou LED diodu
	1.2.1.3.1 The button box
	1.2.1.3.2 Koncová LED dioda
	1.2.1.4Povolení zapnutí
	1.2.2 Přihlášení k přístupu k webové aplikaci
	1.2.2. 1Přístup a přihlášení k rozhraní webové aplikace
	1.2.2. 2Simple understanding of WebApp interface
	1.2.2.2.1Kontrolní oblast
	1.2.2.2. 2Stavový řádek
	1.2. 3Robot parameter setting
	1.2.3.1Nastavení způsobu instalace
	1.2.3. 2Nastavení koncového zatížení
	1 2 3 3 Nastavení souřadnic nástroje
	$12.2.0.0$ Rubba veni souraunie Rubbo ge $1.2.0.0$ 12 4 Robot manual teaching $1^{1}$
	1.2. 10 uční vízuka a záznam vízukových hodů
	1242 Zobrazení informací o výukovém bodu
	1 2 5 Rychlé programování robota
	1.2.5. 1Úvod do jednoduchých pohybových instrukcí
	1 2.5 2Práce s programovými soubory
	1 2 5 3Nansat a snustit nrogram
	13 Manual
	13 1Foreword 2'
	$1.3.11$ What is installed in the hov $2^{\circ}$
	$1.3.1.1$ What is instance in the box $\dots \dots \dots$
	$1.3.1.  2 \text{Important security description} \dots \dots$
	$1.5.1.5  \text{Jak pouzival lulo pillucku} = \frac{1.2.1.4}{1.2.1.4}  \text{Dodyžujto příchučné pouzival}$
	1.2.2 Debet by infinity adjustion
	1.3.2 KODOL DETEL INTRODUCTION $\dots 2.$
	1.3.2. 12akiadni parametry 2
	1.3.2. 2Motion 2
	1.3.2.3Robot coordinate system30
	1.3.2.4Robot Denavit-Hartenbergovy parametry3030

	1.3.2.5Stažení parametrů	DH	33
	1.3.3 Installation		33
1.3.3.1	Pokyny pro zabezpečení		33

	1.3.3.1.1	Stručný úvod		
	1.3.3.1.2	Bezpečnost personálu		
	1.3.3.1.3	Nastavení zabezpečení		
	1.3.3.1.4	Rozpoznání nebezpečí		
	1.3.3.1.5	Informace na výrobním štítku	35	
	1.3.3.1.6	Účinnost a odpovědnost	35	
	1.3.3.1.7	Omezená odpovědnost		
	1.3.3.1.8	Výstražné značky v této příručce		
	1.3.3.1.9	Hodnocení před použitím		
	1.3.3.1.10	Nouzové zastavení		
	1.3.3.1.11	Hnutí Power-Mee		
	40			
	1.3.3.2 Přeprava	a zařízení	41	
	1.3.3.2.1	Doprava	41	
	1.3.3.2.2	Přenášejte		
	1.3.3.2.3	Uložiště	41	
	1.3.3.3 Udržba a	zpracování šrotu	41	
	1.3.3.3.1	Likvidace v rámci údržby	41	
	1.3.3.3.2	Likvidace odpadu robotem		
	1.3.3.4 Specifik	ace instalace	42	
	1.3.31431.3.4	.R1. <sup>01bot</sup> alimmatankatakan kan kan kan kan kan kan kan kan kan		432
	1.3.3.4	4.1.2 Požadavky na instalaci robota M 5	45	
	1.3.3.4	4.1.3 Požadavky na instalaci robota M 10&M16	47	
	1.3.3.4	4.1.4 Požadavky na instalaci robota M 20		
	1.3.3.4.2	Instalace koncovky nástroje		
	1.3.3.4.3	Instalační prostředí	53	
	1.3.3.4.4	Kapacita podlahového nosiče	53	
	1.3.3.4.5	Maximální platné zatížení	53	
	1.3.3.5 Připojer	ní ovládání	53	
	1.3.3.5.1	Rozhraní ovladače	53	
	1.3.3.5.2	Panel I/O řídicí jednotky		
	1.3.3.5.3	Skupina síťových rozhraní RJ45		
	1.3.3.5.4	Koncová deska	59	
	1.3.3.5.5	Pozemek	59	
	1.3.3.5.6	Společné specifikace všech digitálních I/O	61	
	1.3.3.5.7	Bezpečnostní vstupy/výstupy	63	
	1.3.3.5.8	Univerzální digitální množství I/O	65	
	1.3.3.5.9	Digitální vstup M om tlačítko		
	1.3.3.5.10	Interakce s jinými zařízeními nebo PLC	66	
	1.3.3.5.11	Simulace I/O	66	
	1.3.3.6 Demonst	race a ukončení LED	68	
	1.3.3.6.1	Uvod do pole tlačítek	68	
	1.3.3.6.2	MHMI Uvod k výukovému přívěsku	71	
101	1.3.3.6.3	Definice koncove LED		
1.3.4	Rychly start			
	1.3.4.1 Instalace	robotickeho ramene a ovladaciho panelu		
	1.3.4.2 Demonstri 1.3.4.3.1	nae sartup c on, no 1 ro, bo t N ot paired with teach ingpendent		
	1.3.4.3 Buon box	conir orobot movement archingthe oscilloscopy	·····7 8	
	1.3.4.4 Dempern	naster řídí pohyb robota		
1.3.5	Výukový software	pro přívěsky	80	
	1.3.5.1 Základn	í informace	80	
	1.3.5.1.1	Úvod	80	

1.3.5.1.2 Spustit software	80
1.3.5.1.3 Přihlášení uživatele a aktualizace oprávnění	80
1.3.5.2 Počáteční rozhraní systému	81
1.3.5.2.1 Kontrolní oblast	82
1.3.5.2.2 Stavový řádek	83
1.3.5.2.3 Panel nabídek	
1.3.5.2.4 Provozní oblast	86
1353 3D simulace robota	86
1 3 5 3 1 3D virtuální trajektorie a import modelu nástroje	86
1 3 5 3 2 Zohrazani 3D vizualizace oMohotův souřadnicový svetém	00 87
1.2.5.2.2 Nastavaní a zahrazaní znůsobu instalace robota	07 QQ
1354 Robot settings	90
135.4 Robot Sciings	90
1.2542 Souřadnice ovterního nástroje	04
1.3.3.4.2 Sour autice externino hastroje	06
1.2.5.4.4 Dozčířané osové souřadnice	90 07
1.2.5.4.4 KOZSHEILE OSOVE SOUL duffice	97 104
1.5.5.4.5 COMISION IEVEL	104
1.5.5.4.0 MEKKY IIIIII.	104
1.3.5.4./ Koncove zatizeni	105
1.3.5.4.8 Miction compensation	• • • •
107	
1.3.5.4.9 Škálování rychlosti	108
	100
1.3.5.5 Control box I/O	108
13551 Nastavení $I/O$	108
	100
1.3.5.5.2 Zobrazení stavu I/O	109
· · · · · · · · · · · · · · · · · · ·	
1.3.5.5. 3I/O filtrování	110
1.3.5.5. 4I/O configuration	111
1.3.5.6 Robot operation	114
1.3.5.6.1Záznam výukového bodu	114
1.3.5.6. 2Kloubový režim	115
1.3.5.6. 3Báze Jog	116
1.3.5.6. 4Nástroj Jog	117
1.3.5.6. 5Wobj jog	118
1.3.5.6. 6Move	119
1.3.5.6.7 Pohyb osy	120
1.3.5.6. 8TPD (Teach-in programming)	121
1.3.5.7 Teaching simulation	124
1.3.5.7. 1Introduction	124
1 3 5 7 2 Pruh nástroiů	126
1357 3Program command	120
1 3 517 4 r 7 r 4 01 gic (WomhimleacnodmInmtearnfdac e	100
1.5.5 J.5.7.L4. 1510 CW offinitiate action and intrace. C	123
	90
1.3.5.7.4.2if else příkaz	130
	4.04
1.3.5.7.4. 3Goto prikaz	131
1.3.5.7.4.4Prikaz Walt	132
1.3.5.7.4.5Prikaz Pauza	133
12574CD×ilean Dofile	104
1.3.5./.4.b PTIKaZ DOIIIe	134
1.25.747Dřílzoz Vor	195
1.3.3.1.4.1F11Kaz Val	100
 1 3 5 7 5Rozhraní nříkazů pro pohyh	125
$13575$ $10$ $\tilde{r}$ $10$ $\tilde{r}$ $10$ $10$ $10$ $10$ $10$ $10$ $10$ $10$	126
1.0.0.7.7. 1111Ru2 1 11	100
1.3.5.7.5. 2Příkaz Lin	136
	100
· · · · · · · · · · · · · · · · · · ·	

1.3.5.7.5.7	Příkaz Spline	140
1.3.5.7.5.8	Příkaz N-Spline	141
1.3.5.7.5.9	Příkaz Weave	141
1.3.5.7.5.10	Příkaz TPD	142
1.3.5.7.5.11	Příkaz Offset	
1.3.5.7.5.12	Příkaz ServoCart	144
1.3.5.7.5.13	Příkaz Trajctory	145
1.3.5.7.5.14	Příkaz TrajctoryJ	146
1.3.5.7.5.15	Příkaz DMP	147
1.3.5.7.5.16	Příkaz WPTrsf	147
1.3.5.7.6 Rozhra	aní řídicích příkazů	
1.3.5.7.6.1	Příkaz IO	
1.3.5.7.6.2	Příkaz AI	
1.3.5.7.6.3	Příkaz Vir-IO	
1.3.5.7.6.4	Příkaz Aux-IO	
1.3.5.7.6.5	Příkaz MoveDO	
1.3.5.7.6.6	Příkaz ToolList	
1.3.5.7.6.7	Příkaz Mode	
1.3.5.7.6.8	Příkaz ke kolizi	
1.3.5.7.6.9	Příkaz Acc	
1.3.5.7.7 Perifer	ni prikazove rozhrani	
1.3.5.7.7.1	Prikaz Gripper	
1.3.5.7.7.2	Prikaz Spray	
1.3.5.7.7.3	Prikaz EAxis	
1.3.5.7.7.4	Prikaz Convey	
1.3.5.7.8 Rozhra	ini prikazu pro svarovani	
1.3.5.7.8.1	Prikaz ke svarovani	
1.3.5.7.8.2	Prikaz Segment	
1.3.5.7.8.3	Laserovy prikaz	
1.3.5.7.8.4	Prikaz LI-Rec	
1.3.5.7.8.5	Prikaz w-Search	
1.3.3.7.8.0	Prikdz Welu-Irc	
1.3.3.7.8.7	Prikaz Aujusi	
1.5.5.7.9 ROZIIId	III prikazu Force Control	104 164
1.3.3.7.9.1	PTIKAZ F/1	104 165
1.3.3.7.9.4 1.2 E 7 10 Viguól	PTIKAZ KLOULICIILO IILOILLEILU	103 165
1.5.5.7.10 VIZUAL	111 prika20ve rozni ani	103
1.3.3.7.10.1 1.2 = 7.11 Dřílzoz	SD prikaz	100
1.5.5.7.11 PIIKdZ	Dříkaz pro paletu	100
1.3.3.7.11.1 1.2.5.7.12 Komur	pilkaz pro paletu nikační příkazová rozhraní	107 167
1 2 5 7 1 7 1	Dřílaz Modhus	160
1 2 5 7 19 9	Příkaz Ymlune	160
135712 Domos	né příkazové rozbraní	109 170
1257121	Příkaz vlákna	170 171
1 3 5 7 1 3 9	Funkční nříkaz	1/1 171
135714 Kódow	ání výukového programu	
1.3.5.7.15 Místní	výukový hod	174
1.3.5.7.16 Aktuáli	ní záloha programu	176
1.3.5.7.17 Grafiel	ké programování	177
1.3.5.7 17 1	Grafické příkazy pro programování pohybu	178
1.3.5.7.17.1	Ovládání grafických programovacích příkazů	178
1357173	Pokročilé grafické programovací příkazy	179
1 2 5 7 17 /	Příklad noužití grafických programovacích příkazů	120
1.3.3.7.17.4	י דוגומע אסטבונו ברמוונגינוו ארטבו מווטימנונוו אווגמבע	

1.3.5.7.17.5 Modularizace bloků grafického programového kódu .	180
1.3.5.7.18 Rízení výuky	182
1.3.5.8 Status information	184
1.3.5.8.1 Systémový protokol	184
1.3.5.8. 2Dotaz na stav	185
1.3.5.9 Auxiliary application	186
1359 1Aktualizace systému	186
1.35.9 2Data hackup	188
1350 310c datový záznam	120
1.250.4 Touch point configuration	105
	190
1.5.5.9. $OPTESUIT IIIduce$	190
1.3.5.9. 6Puvoa prace	192
1.3.5.9. /Interference zone configuration	193
1.3.5.9. 8Terminal LED configuration	196
1.3.5.9.9Peripheral protocol	197
1.3.5.9.10 Konfigurace hlavního programu	199
1.3.5.9.11 Zámek přetažení	199
1.3.5.9.12 Knihovna odborníků na svařování	200
1.3.5.9.13 Bezpečné nastavení rychlosti	206
1.3.5.9.14 Security Wall Configuration	207
1.3.5.9.15 Bezpečnostní démon	209
1 3 5 10 Nastavení systému	 209
1.35101 Obecná nastavaní	203
1 2 5 10 1 1 Nastavoní sítě	205
	210
1 3 5 10 2 Nastavení účtu Nastavení účtu	210
1.5.5.10.2 Wastaveni učta Wastaveni učta Mastaveni učta	210
1.5.5.10.2 $2$ Autority management	210
1.2.5.10.2. ZAUIOTITY IIIdildgenient	213
1.5.5.10.2. $3 IIIII port/Export$	217
1.0.5.10.5 U	210
	218
1.3.5.10.4.1Konfigurace rozsahu parametru	219
1.3.6 Peripheral	220
1.3.6.1 Konfigurace periferií chapadla	220
1.3.6.1.1Kroky vyuky programu Gripper	220
1.3.6.1.2Vyuka programu Gripper	222
1.3.6.2 Konfigurace periferil stříkací pistole	222
1.0.0.0.1Knolwy konfigurace newiferiú	 111
1.5.6.2.1Kroky konnigurace pernerni surikaci pistole	222
1.3.6.2.2 Vyuka programu sprejovani	223
1.3.6.3Konfigurace periferil svarecky	223
	· · · • • • • • • • • • • • • • • • • •
1.3.6.3.1KONIIgurace periferil svarecky	223
1.3.6.3. Zweiding program teaching	224
1.3.6.4 Konfigurace perifernino zarizeni senzoru	225
1.2.6.4.1 Knolwy konfigurace new ferrie concerns	 חחר
1.5.0.4.1Kroky konnigurace periferie senzoru	225
1.3.6.4. ZLaser sensor tracking function	229
1.3.6.4.3Funkce reprodukce trajektorie laserového snímače	231
1.3.6.5Rozšířená konfigurace periferních os	232
1.3.6.5.1Kroky konfigurace rozšířených periferií osv	232

1.3.6.5.2 Roz	šířená osa s výukovým programem j	pro svařování s laserovým sledováním .	235
1.3.6.6 Konfigu	race sledování dopravníku	·····	236
1.3.6.6 1.3.6.6.2Výu 1.3.6.7 l	.1 Kroky konfigurace sledování o ikový program sledování dop Konfigurace adaptace na výšku 1.3.6.7.1Kroky konfigurace ada	lopravníku ravníkového pásu	236 240 240 240

	1.3.6.7.2	2 Samočinná adaptace na výšku s rozšířenou osou a laserovým sledováním svařo	vání učit-
		program	241
	1.3.6.8 Konfi	gurace periferie snímače síly a točivého momentu	243
	1.3.6.8.2	I Kroky konfigurace snímače síly a točivého momentu	243
	1.3.6.8.2	2 Identifikace zatížení snímače síly a točivého momentu	244
	1.3.6.8.3	3 Snímač síly a točivého momentu s asistencí při přetahování	244
	1.3.6.8.4	4 Snímač síly/točivého momentu Detekce kolize	247
	1.3.6.8.5	5 Snímač síly/točivého momentu Řízení síly Pohyb	248
	1.3.6.8.0	G Zavedení šroubu snímače síly/točivého momentu	249
	1.3.6.8.7	7 Snímač síly/motoru Otočné vkládání	250
	1.3.6.8.8	3 Snímač síly/motoru Přímé vložení	251
	1.3.6.8.9	9 Snímač síly/motoru Orientace povrchu	252
	1.3.6.8.2	10 Centrování snímače síly/motoru	253
	1.3.6.8.1	11 Snímač síly/točivého momentu Detekce síly poklepu	254
	1.3.6.9 Rozší	řená konfigurace periferních zařízení IO	254
	1.3.6.9.1	l Rozšířené kroky konfigurace zařízení IO	254
	1.3.6.10 Konfi	gurace paletizačního systému	255
	1.3.6.10	.1 Kroky konfigurace paletizačního systému	255
1.3.7	Příloha		259
	1.3.7.1 Doda	tek 1: Chyby regulátoru pohybu a způsoby jejich řešení	259
	1.3.7.2 Přílol	na 2: Tabulka chybových kódů ovladače serva	261
	1.3.7.3 Doda	tek 3: Koncová deska 485 upgrade	266
	1.3.7.4 Doda	tek 4: Upgrade řídicí jednotky 485	269
	1.3.7.5 Doda	tek 5: Seznam náhradních dílů a zranitelných dílů	269
1.3.8	Termín	·	270
	12		272
2 Prirucka SD	N.		2/3 272
2.1 CTT.	c	nagifikaga datavé etmiktury	273 979
2.1.1	ე1117-ლირი ე1117-ლირი	pecifikace uatove struktury	213
	2.1.1.11yp nav		213
	2.1	1 2Datový tvn polohy kloubu	273
	2.1.1 3Datov	ý typ prostorové nolohy	274
	2.1.1.0 D uto 1	14 Fuler Angle attitude data type	274
	<b>د</b> .بے	2.1.1 5Cartesian snace nose data type	274
	2	11 6Extension axis position data type	274
	_	2 1 1 7Torque sensor data tune	275
	2 1 1 8 Datowý t	2.1.1. 7101que sensor unu type	Snirála
212	Ziiiio Datovy t	yp parametra a	275
2.1.2	2121 Insta	nciovat robota	275
	2.1.2.1 motal	že komunikaci s řídicí jednotkou	276
	2.1.2.2 Hava	z na číslo verze SDK	276
	2.1.2.5 Dotaz 2.1.2.5 Dotaz	ní ID adresu řadiče	276
	2.1.2.4 213ku 2.1.2.5 Ovláu	lání rohota pro vstup do režimu výuky tažením neho výstup z něj	276
	2.1.2.5 Oviat	zuje se zda je rohot v režimu přetahování	276
	2.1.2.0 Dotaz	ení ovládání nahoru a dolů	270 977
	2.1.2.7 Povol 2.1.2.8 Ruční	lantomatický režim ovládání rohota	277
	2.1.2.0 Ruch 2.1.2.0 Příkl	ad kódu	277
213	Pohyh	uu 1.0uu	278
2.1.3	2131 Pohy	h hodu Iog	278
	2.1.3.1 Torry 2.1.3.2 7actor	vení dvnamického znomalení v hodě Iog	278
	2133 Rěho	se okamžitě zastaví	279
	2.1.3.5 Define 2.1.3.5 Define 2.1.3.6 Define	ad kódu	279
	2.1.3.7 FIIM 2125 Dobu	h v klouhním prostoru	<u>2</u> 75 281
	2.1.3.3 FULLY 2.1.2.6 Dřím	očarý nahuh v kartázekám prostoru	201 701
	2.1.3.0 FIIIII	א אמונפגאבווו או אפור איז א אמו א אמו איז א א	201

	2.1.3.	7Kruhový obloukový pohyb v kartézském prostoru	282
	 0 1 0	OV myhovyý v obyh v kontázaltán prostomy	റററ
	2.1.3.	Opythed had	283
	2.1.3.	Spirálový pohyby kartázskám prostoru	203
	2.1.3.103	Spiratovy potryb v Kartezskeni prostoru	200
	2.1.3.11	Priklau kouu Debyby z režimu comronebenu ze eneležném prestoru	280
	2.1.3.12	Poliyb v rezimu servopononu ve společnem prostoru	207
	21313	Příklad kódu	287
	2.1.3.14	Cartesian space servo mode motion	288
	2 1 3 15	Příklad kódu	289
	2.1.3.16	Pohyb z bodu do bodu v kartézském prostoru	290
	2.1.3.17	Příklad kódu	290
	2.1.3.18	The spline motion begins	291
	2.1.3.19	Spline motion PTP	292
	2.1.3.20	The spline movement ends	292
	2.1.3.21	Příklad kódu	292
	2.1.3.22	Ukončovací pohyb	294
	2.1.3.23	The whole point shift begins	294
	2.1.3.24	The whole point shift ends	294
	2.1.3.25	Příklad kódu	294
2.1.4	IO		296
	2.1.4.	1Set the control box digital output	296
	2.1.4.	2Nastavení digitálního výstupu nástroje	296
	2.1.4.	3Nastavení analogového výstupu řídicí jednotky	297
	2.1.4.	4Nastavení analogového výstupu nástroje	297
	214	5Get the control box digital input	297
	2.1.4.	6Získání číselného vstupu nástroje	297
	2.1.4.	7Wait for the control box digital input	298
	2.1.4.	8Čekání na digitální vstup multiplexu řídicí skříně	298
	2.1.4.9Po	čkejte na zadání čísla nástroje	298
	2.1.4.10	Get control box analog input	299
	2.1.4.11	Získání analogového vstupu nástroje	299
	2.1.4.12	Čekání na analogový vstup řídicí jednotky	299
	21/12	Čekání na analogový vetup nástroje	200
	2.1.4.13		299
	2.1.4.14	Příklad kódu	300
2.1.5	Comm	on Settings	302
	2.1.5.1	Set global speed	302
	2.1.5.2	Nastavení hodnoty systémové proměnné	302
	2.1.5.3	Set tool coordinate system	302
	2.1.5.4	Set the tool coordinate list	302
	2.1.5.5	Nastavení vnějšího souřadnicového systému nástroje	303
	2.1.5.6	Nastavení seznamu externích souřadnicových systémů nástroje	303
	2.1.5.7	Nastavení souřadnicového systému obrobku	303
	2.1.5.8	Set the list of work coordinate systems	303
	2.1.5.9	Nastavte hmotnost koncového zatížení.	304
	2.1.5.10	Nastavení souřadnic centroidu koncového zatížení	304

	2.1.5.11	Set the robot installation mode	304
	2.1.5.12	Set the robot installation Angle	304
	2.1.5.13	Wait for the specified time	304
	2.1.5.14	Code example	305
2.1.6	Zabezpe	settings	307
	čení		
	2.1.6.1	Set collision level	307
	2.1.6.2	Nastavení zásad po kolizi	307
	2.1.6.3	Nastavte kladnou mez.	307
	2.1.6.4	Nastavte zápornou mez.	307
		<b>▲</b>	

		2.1.6.5	Vymazání chybového stavu	308
		2.1.6.6 2.1.6.7	Společný přepínač kompenzace	
		308 2.1.6.8	Nastavení koeficientu kompenzace kloubu M - boční montáž	
		2.1.6.9	Nastavení koeficientu kompenzace kloubu M - zpětná montáž	
		2.1.6.10	Nastavení společného koeficientu kompenzace M - M ee mount	
		2.1.6.11	Příklad kódu	309
	2.1.7	Stavový	dotaz	
		2.1.7.1	Získání úhlu montáže robota	310
		2.1.7.2	Získání hodnoty systémové proměnné	310
		2.1.7.3	Získání aktuální polohy kloubu (úhel)	310
		2.1.7.4	Získání aktuální polohy kloubu (v radiánech)	311
		2.1.7.5	Získat aktuální polohu nástroje	311
		2.1.7.6	Získání aktuálního čísla souřadného systému nástroje	311
		2.1.7.7	Získání aktuálního čísla souřadnicového systému obrobku	311
		2.1.7.8	Získání aktuální polohy koncové příruby	312
		2.1.7.9	Řešení inverzní kinematiky	312
		2.1.7.10	Řešení inverzní kinematiky	312
		2.1.7.11	Řešení inverzní kinematiky	312
		2.1.7.12	Řešení dopředné kinematiky	313
		2.1.7.13	Zjištění aktuálního kloubního momentu	313
		2.1.7.14	Zjištění hmotnosti aktuálního nákladu	313
		2.1.7.15	Zjištění středu hmotnosti aktuálního nákladu	313
		2.1.7.16	Získání aktuálního souřadnicového systému nástroje	314
		2.1.7.17	Get the current work M ame 314	
		2.1.7.18	Získat měkký mezní úhel kloubu	314
		2.1.7.19	Získat systémový čas	314
		2.1.7.20	Získat aktuální konfiguraci kloubů robota	315
		2.1.7.21	Získání aktuální rychlosti	315
		2.1.7.22	Dotaz, zda je pohyb robota dokončen	315
		2.1.7.23	Příklad kódu	315
	2.1.8	Opakov	vání trajektorie	317
		2.1.8.1	Nastavení parametrů nahrávání stopy	317
		2.1.8.2	Spuštění nahrávání stopy	318
		2.1.8.3	Zastavení nahrávání stopy	318
		2.1.8.4	Vymazat záznam	318
		2.1.8.5	Příklad kódu	318
		2.1.8.6	Přednačítání trajektorie	319
		2.1.8.7	Opakování trajektorie	319
		2.1.8.8	Příklad kódu	320
	2.1.9	Použití p	programu WebAPP	321
		2.1.9.1	Nastavení automatického načítání výchozího programu úloh při spuštění	321
		2.1.9.2	Načtení zadaného programu úlohy	321
		2.1.9.3	Získat název načteného programu úlohy	321
		2.1.9.4	Získání čísla řádku aktuálního programu úlohy robota	321
		2.1.9.5	Spustit aktuálně načtený program úlohy	
		2.1.9.6	Pozastaveni aktuálné spušténého programu úlohy	
		2.1.9.7	Obnoveni aktuálně pozastaveného pracovního programu	
		2.1.9.8	ukonci aktuálné spušténý program úlohy	
		2.1.9.9	Ziskani stavu provádění programu úlohy robota	
		2.1.9.10	Priklad kódu	323
	2.1.10	Perifer	nı	324
viii		2.1.10.1	Konfigurace chapadla	<u>324</u>
		2.1.10.2	Ziskani konfigurace chapadla	324

. . . . . . . . . . .

		2.1.10.3	Aktivace chapadl	a	324
		2.1.10.4			325
		2.1.10.5	lískání stavu pohy	ybu chapadla	325
		2.1.10.6	Příklad kódu		325
	2.1.11	Kontrol	síly		326
		2.1.11.1	Konfigurace sníma	ače síly	326
		2.1.11.2	Lískání konfigura	ce snímače síly	326
		2.1.11.3	Aktivace snímače	síly	327
		2.1.11.4	Kalibrace snímače	e síly	327
		2.1.11.5	Příklad kódu		327
		2.1.11.6	Vastavení referen	ičního souřadného systému snímače síly	328
		2.1.11.7	láznam o identifi	kaci hmotnosti nákladu	329
		2.1.11.8	/ýpočet identifika	ice hmotnosti nákladu	329
		2.1.11.9	Vačtení identifika	čního záznamu centroidu	329
		2.1.11.10	/ýpočet identifika	ice centroidu zatížení	329
		2.1.11.11	Příklad kódu		329
		2.1.11.12	lískání údajů o síle	e/točivém momentu v referenčním souřadném systému	331
		2.1.11.13	lískání nezpracov	raných údajů o síle/otáčivém momentu ze snímače síly	331
		2.1.11.14	)chranný systér	n proti nárazu	331
		2.1.11.15	?říklad kódu		332
		2.1.11.16	Rízení konstantní	síly	333
		2.1.11.17	Příklad kódu		333
		2.1.11.18	spirálový průzku	ım	335
		2.1.11.19	<pre></pre>		335
		2.1.11.20	lineární vkládán	ú	335
		2.1.11.21	Příklad kódu		336
		2.1.11.22	Polohování na po	vrchu	338
		2.1.11.23	/ýpočet polohy st	ředové roviny začíná	338
		2.1.11.24	/ýpočet konců po!	lohy ve střední rovině	338
		2.1.11.25	Příklad kódu		338
		2.1.11.26	Kompatibilní kon	trola na	340
		2.1.11.27	/ypnutá kontrola	shody	340
		2.1.11.28	Příklad kódu		340
2.2	Pytho	n	,		
	2.2.1	Základi			
		2.2.1.1	nstancování robo	, , , , , , , , , , , , , , , , , , ,	
		2.	1.1.1 Příklad	kódu	
		2.2.1.2	Lislo verze dotaz	u SDK	343
		2.	1.2.1 Příklad	kódu	343
		2.2.1.3	Liskání IP adresy ř	adiće	
		2.	1.3.1 Priklad	kódu	
		2.2.1.4	repinač manuální	ho/automatického režimu řídicího robota	
		2.	1.4.1 Priklad	kódu	
		2.2.1.5	Režim přetahován	ii robota	
		2.	1.5.1 Ovládár	ií robota pro vstup do režimu výuky tažením nebo výstup z něj	
		2.	1.5.2 Zkontro	lujte, zda je robot v rezimu pretahovani	
		0.0.1.0	2.2.1.5.2.1	Priklad kodu	
		2.2.1.6	Jvladani robota p	pro povoleni nebo sniženi povoleni	
	0 0 0	2.	1.6.1 Priklad	kodu	
	2.2.2	Pohyb.			
		2.2.2.1	kobot Jog		
		2.	2.1.1 Jog Jog.		
		2.	2.1.2 Jog Jog Z	pomaleni zastavi	346
		2.	.2.1.3 jog jog s	e okamzite zastavi	347
					<b>_</b>
			-2.2.2.1.3.1	Priklad kodu	
					IX.

	2.2.2.2 Pohyb v kloubním prostoru	349
	2.2.2.2.1 Příklad kódu	
	2.2.2.3 Lineární pohyb v kartézském prostoru	351
	2.2.2.3.1 Příklad kódu	351
	2.2.2.4 Pohyb po kruhovém oblouku v kartézském prostoru	352
	2.2.2.4.1 Příklad kódu	352
	2.2.2.5 Kruhový pohyb v kartézském prostoru	353
	2.2.2.5.1 Příklad kódu	
	2.2.2.6 Spirálový pohyb v kartézském prostoru	354
	2.2.2.6.1 Příklad kódu	355
	2.2.2.7 Pohyb v kloubním prostoru v režimu serva	355
	2.2.2.7.1 Příklad kódu	355
	2.2.2.8 Pohyh v režimu serva v kartézském prostoru	356
	2.2.2.8.1 Příklad kódu	356
	2.2.2.9 Pohyh z hodu do hodu v kartézském prostoru	357
	2.2.2.91 Příklad kódu	357
	2.2.2.5.1 Printed Road	358
	2.2.2.10 Robot spine motion start	358
	2.2.2.10.1 Spine noton start	358
	2.2.2.10.2 Drazkovy ponyb 111	358
	2.2.2.10.5 Konec ponybu urazkovani	
	2.2.2.10.3.1 FIRIAU KOUU	350
	2.2.2.11 KODOLINEW Splitte MOUOI	250
	2.2.2.11.1 Nový začatek polybu splite	250
	2.2.2.11.2 Nové ul d2Kové 2dKolicelli poliybu	200
	2.2.2.11.5 Nove bouy pokyila spille	200
	2.2.2.2.11.5.1 PTIKIdu KUUU	
	2.2.2.12 RODOL UKONCI PONYD	
	2.2.2.12.1 PTIKIdu Kouu	
	2.2.2.13 CEIKOVY POSULI OMODOL DOUU	
	2.2.2.13.1 Výchozi bou čelkový posuli	
	2.2.2.13.2 Celkovy posuli kolicu dodu	
0 0 0	2.2.2.13.2.1 Priklad kodu	
2.2.3	IU	
	2.2.3.1 Nastaveni digitalnino vystupu ridici jednotky	
	2.2.3.1.1 PTIKIAO KOOU	
	2.2.3.2 Nastaveni digitamino vystupu nastroje	
	2.2.3.3 Nastaveni analogoveho vystupu ridici jednotky	
	2.2.3.3.1 Priklad kodu	
	2.2.3.4 Nastaveni analogoveno vystupu nastroje	
	2.2.3.4.1 Priklad Kodu	
	2.2.3.5 Liskejte digitalni vstup ridici jednotky	
	2.2.3.5.1 Priklad Kodu.	
	2.2.3.6 Ziskani digitalnino vstupu nastroje	
	2.2.3.6.1 Priklad kodu.	
	2.2.3.7 Cekani na digitalni vstup M om ovladaci skrinka	) 
	2.2.3.7.1 Priklau Kouu	
	2.2.3.0 CERAIII Ha VICE UIGHAIHICH VSUIPU IN OIL OVIAUACI SKEIIIKa	267
	2.2.3.0.1 FIIKau Kouu	102 267
	2.2.3.3 CERAILI Ha UISHAILI VSHUP HASH OJE	260
	2,2,3,3,1 FIIKIau Kouu	200
	2.2.3.10 CERAILI HA UIGHAILI VSHUP LETHILIAIU.	
	2.2.3.10.1 PTIKIAU KUUU	
		0.00
	2.2.3.11 LISKANI ANAIOGOVENO VSLUPU NASIFOJE	
	2.2.3.12 Cekani na zadani simulace ridici jednotky	

	<ul> <li>2.2.5.2.1 Příklad kódu.</li> <li>2.2.5.3 Nastavení kladného limitu.</li> <li>2.2.5.3.1 Příklad kódu.</li> <li>2.2.5.4 Nastavení záporného limitu.</li> <li>2.2.5.4.1 Příklad kódu.</li> <li>2.2.5.5 Vymazání chybového stavu.</li> <li>2.2.5.5.1 Příklad kódu.</li> <li>2.2.5.6 Společný přepínač kompenzace</li></ul>	
	<ul> <li>2.2.5.2.1 Příklad kódu.</li> <li>2.2.5.3 Nastavení kladného limitu.</li> <li>2.2.5.3.1 Příklad kódu.</li> <li>2.2.5.4 Nastavení záporného limitu .</li> <li>2.2.5.4.1 Příklad kódu.</li> <li>2.2.5.5 Vymazání chybového stavu.</li> <li>2.2.5.5 Vymazání chybového stavu.</li> <li>2.2.5.6 Společný přepínač kompenzace</li></ul>	378 378 378 378 378 379 379 379 380
	<ul> <li>2.2.5.2.1 Příklad kódu.</li> <li>2.2.5.3 Nastavení kladného limitu.</li> <li>2.2.5.3.1 Příklad kódu.</li> <li>2.2.5.4 Nastavení záporného limitu .</li> <li>2.2.5.4 Příklad kódu.</li> <li>2.2.5.5 Vymazání chybového stavu.</li> <li>2.2.5.5 Vymazání chybového stavu.</li> <li>2.2.5.5.1 Příklad kódu.</li> <li>2.2.5.6 Společný přepínač kompenzace</li></ul>	377 378 378 378 378 379 379 379 379 380
	<ul> <li>2.2.5.2.1 Příklad kódu.</li> <li>2.2.5.3 Nastavení kladného limitu.</li> <li>2.2.5.3.1 Příklad kódu.</li> <li>2.2.5.4 Nastavení záporného limitu .</li> <li>2.2.5.4.1 Příklad kódu.</li> <li>2.2.5.5 Vymazání chybového stavu.</li> <li>2.2.5.5.1 Příklad kódu.</li> <li>2.2.5.6 Společný přepínač kompenzace</li></ul>	377 378 378 378 378 379 379 379 379
	<ul> <li>2.2.5.2.1 Příklad kódu.</li> <li>2.2.5.3 Nastavení kladného limitu.</li> <li>2.2.5.3.1 Příklad kódu.</li> <li>2.2.5.4 Nastavení záporného limitu.</li> <li>2.2.5.4.1 Příklad kódu.</li> <li>2.2.5.5 Vymazání chybového stavu.</li> <li>2.2.5.5.1 Příklad kódu.</li> <li>2.2.5.6 Společný přepínač kompenzace</li></ul>	378 378 378 378 378 379 379 379 379
	2.2.5.2.1       Příklad kódu         2.2.5.3       Nastavení kladného limitu         2.2.5.3.1       Příklad kódu         2.2.5.4       Nastavení záporného limitu         2.2.5.4.1       Příklad kódu         2.2.5.5       Vymazání chybového stavu         2.2.5.5.1       Příklad kódu         2.2.5.6       Společný přepínač kompenzace         379       2.2.5.6.1         Příklad kódu	377 378 378 378 378 379 379 379 379
	2.2.5.2.1       Příklad kódu         2.2.5.3       Nastavení kladného limitu         2.2.5.3.1       Příklad kódu         2.2.5.4       Nastavení záporného limitu         2.2.5.4.1       Příklad kódu         2.2.5.5       Vymazání chybového stavu         2.2.5.5.1       Příklad kódu         2.2.5.5.1       Příklad kódu         2.2.5.5.1       Příklad kódu	378 378 378 378 378 379 379 379
	2.2.5.2.1       Příklad kódu         2.2.5.3       Nastavení kladného limitu         2.2.5.3.1       Příklad kódu         2.2.5.4       Nastavení záporného limitu         2.2.5.4.1       Příklad kódu         2.2.5.5       Vymazání chybového stavu	378 378 378 378 378 379 379
	2.2.5.2.1       Příklad kódu         2.2.5.3       Nastavení kladného limitu         2.2.5.3.1       Příklad kódu         2.2.5.4       Nastavení záporného limitu         2.2.5.4.1       Příklad kódu	378 378 378 378 378 379
	2.2.5.2.1       Příklad kódu         2.2.5.3       Nastavení kladného limitu         2.2.5.3.1       Příklad kódu         2.2.5.4       Nastavení záporného limitu	378 378 378 378 378
	2.2.5.2.1 Příklad kódu 2.2.5.3 Nastavení kladného limitu 2.2.5.3.1 Příklad kódu	378 378 378
	2.2.5.2.1       Příklad kódu         2.2.5.3       Nastavení kladného limitu	378 378
	2.2.5.2.1 Příklad kódu	
	2.2.5.2 Nastavení strategie po kolizi	277
	2.2.5.1 2.25ettip-le	377
2.2.5	Nastavení zabezpečení	377
	2.2.4.13.1 Příklad kódu	377
	2.2.4.13 Cekání na zadaný čas	376
	2.2.4.12.1 Příklad kódu	376
	2.2.4.12 Nastavení souřadnic centroidu koncového zatížení	376
	2.2.4.11.1 Příklad kódu	376
	2.2.4.11 Nastavení úhlu instalace robota - M ee instalace	
	2.2.4.10.1 Příklad kódu	375
	2.2.4.10 Nastavení způsobu instalace robota - pevná instalace	375
	2.2.4.9.1 Příklad kódu	375
	2.2.4.9 Nastavení hmotnosti koncového zatížení	374
	2.2.4.8.1 Příklad kódu	
	2.2.4.8 Nastavení tabulky souřadnicových řad obrobků	374
	2.2.4.7.1 Příklad kódu	374
	2.2.4.7 Nastavení souřadnicového systému obrobku	373
	2.2.1.0 russaven tabuty souraune externet hastroju	373
	2.2.4.5.1 FIIKau Kouu	
	2.2.4.5 INASIAVEIII EXTERIIIIO SOURAUIICOVEIIO SYSTEMU NASTROJE	372 272
	2.2.4.4.1 PTIKIAU KOUU	312 272
	2.2.4.4 Tabulka souradnicovych rad nastroju	
	2.2.4.3.1 Příklad kódu	372
	2.2.4.3 Nastavení souřadnicového systému nástroje	
	2.2.4.2.1 Příklad kódu	
	2.2.4.2 Nastavení hodnot systémových proměnných	371
	2.2.4.1.1 Příklad kódu	371
	2.2.4.1 Nastavení globální rychlosti	370
2.2.4	Společná nastavení	370
	2.2.3.13.1 Příklad kódu	370
	2.2.3.13 Čekání na analogový vstup nástroje	370

2.2.6	Stavový dotaz	
	2.2.6.1 Získání úhlu instalace robota	
	2.2.6.1.1 Příklad kódu	
	2.2.6.2 Získání hodnot systémových proměnných	
	2.2.6.2.1 Příklad kódu	
	2.2.6.3 Zjištění aktuální polohy kloubu (úhel)	
	2.2.6.3.1 Příklad kódu	
	2.2.6.4 Získání aktuální polohy kloubu (radián)	383
	2.2.6.4.1 Příklad kódu	384
	2.2.6.5 Získání aktuální polohy nástroje	384
	2.2.6.5.1 Příklad kódu	384
	2.2.6.6 Získání aktuálního čísla souřadnicového systému nástroje	384
	2.2.6.6.1 Příklad kódu	385
	2.2.6.7 7ískání aktuálního čísla souřadnicového systému obrohku	385
	2.2.0.7 Ziskuli uktualililo tola souradileoveno systemu obrosku	
	2.2.0.7.1 THRidu Roud	
	2.2.0.0 <i>L</i> jisteni aktualni polony koncove priraby	
	2.2.0.0.1 FIIKlau Kouu	
	2.2.0.5 RESEIL HIVELAIL KIICHII ALIKY	
	2.2.0.7.1 PIIKldu Kouu	
	2.2.0.10 Resetti ilivetziti killentatiky - Zauatti referenciti pototty	
	2.2.0.10.1 Priklau kouu	/ ۵۵ ۲۵۵
	2.2.6.11 Resemi inverzni kinematiky - zua existuje resemi	
	2.2.6.11.1  Prikiad kodu	
	2.2.6.12 Resent dopredne kinematiky	
	2.2.6.12.1 PTIKIAO KOOU.	
	2.2.6.13.1 Priklad kodu	
	2.2.6.14 Zjisteni hmotnosti aktualniho nakladu	
	2.2.6.14.1 Priklad kodu	
	2.2.6.15 Ziskáni centroidu aktuálního zatižení.	
	2.2.6.15.1 Příklad kódu	
	2.2.6.16 Získání aktuálního souřadnicového systému nástroje	
	2.2.6.16.1 Příklad kódu	
	2.2.6.17 Získání aktuálního souřadnicového systému obrobku	
	2.2.6.17.1 Příklad kódu	
	2.2.6.18 Získat měkký mezní úhel kloubu	
	2.2.6.18.1 Příklad kódu	
	2.2.6.19 Získat systémový čas	
	2.2.6.19.1 Příklad kódu	
	2.2.6.20 Zjištění aktuální konfigurace kloubů robota	
	2.2.6.20.1 Příklad kódu	
	2.2.6.21 Získat výchozí rychlost	
	2.2.6.21.1 Příklad kódu	
	2.2.6.22 Zkontrolujte, zda je pohyb robota dokončen	
	2.2.6.22.1 Příklad kódu	
2.2.7	Opakování trajektorie	
	2.2.7.1 Nastavení parametrů záznamu trajektorie	
	2.2.7.1.1 Příklad kódu	
	2.2.7.2 Spuštění záznamu trajektorie	
	2.2.7.3 Zastavení záznamu trajektorie	
	2.2.7.3.1 Příklad kódu	
	2.2.7.4 Odstranění záznamu trajektorie	
	2.2.7.4.1 Příklad kódu	396
	2.2.7.5 Přednačítání trajektorie	396

	2.2.7.6 Reprodukce trajektorie	396
	2.2.7.6. 1Příklad kódu	396
2.2	2.8 WebAPP program use	397
	2.2.8.1 Nastavení a automatické načtení výchozího operačního programu	397
	 2 2 8 1 1Dříklad kódu	307
	2.2.8.1. IFINIAU KOUU	397
	2.2.0.2 Nuclein Zudaneno programa alony	397
	2.2.8.2. 7ijštění čísla řádku provádění aktuálního programu úloby robota	398
	2.2.8.8 2 Spiseri elsa racka provacen aktualino programa dony robota	398
	2.2.8. 5Pozastavení aktuálně spuštěného programu úlohy	398
	2.2.8.60bnovení aktuálně pozastaveného pracovního programu	398
		200
	2.2.8. / UKOIICII prave spusieny program ulony	399
		399
	2.2.8.9Zjistěte název načteného programu úlohy	399
	2.2.8.9. 1Příklad kódu	399
2.2	2.9 Peripheral	400
	2.2.9.1 Obtain gripper configuration	400
	2.2.9. 2Aktivovat chapadlo	400
	2.2.9.30vládací chapadlo	401
	2.2.9. 4Zjištění stavu pohybu chapadla	401
	2.2.9.5Konfigurace chapadla	401
	2.2.9.5. 1Příklad kódu	402
2.2	2.10 Force control	402
	2.2.10.1 Získání konfigurace snímače sílv	402
	2.2.10.2 Konfigurace snímače síly	402
	2.2.10.2.1 Příklad kódu	403
	2.2.10.3 Aktivace snímače síly	403
	2.2.10.3.1 Příklad kódu	403
	2.2.10.4 Zero calibration of force sensor	403
	2.2.10.4.1 Příklad kódu	404
	2.2.10.5 Set the force sensor reference coordinate system	404
	2.2.10.5.1 Příklad kódu	404
	2.2.10.6 Výpočet identifikace hmotnosti nákladu	404
	2 2 10 7 Záznam o identifikaci hmotnosti nákladu	405
	2.2.10.7.1 Příklad kódu	405
	2.2.10.8 Load centroid identification calculation	405
	2.2.10.9 Load centroid identification record	406
	2.2.10.9.1 Příklad kódu	406
	2.2.10.10 Získání údajů o síle/motoru v referenčním souřadnicovém systému	406
		407
	2.2.10.10.1 Priklad Kodu 2.2.10.11 Získání nezpracovaných údajů o síle/otáčivém momentu M ze snímače sílv	407
		y
	407	407
	2.2.10.11.1 FIIKidu Kouu	407
	2.2.10.12 OUII alia pi ou Kolizi	407
	2.2.10.12.1 FIIKiau Kouu 2.2.10.12 Ďízoní konstantní síly	400 //10
	2.2.10.13 M2CH KONSTAND SHY	400 ///0
	2 2 10 14 Průzkum snirálovitých linií	409 409
	2 2 10 14 1 Příklad kódu	410
	2.2.10.15 Vložení rotace	410

• • • • • • • • • • • • • • • • • • • •	
2.2.10.15.1 Příklad kódu	411
2.2.10.16 Lineární vkládání	412
2.2.10.16.1 Příklad kódu	412
2.2.10.17 Calculate the middle plane position to start	413
2.2.10.18 Calculate the middle plane position to end	413

	2.2.10.19 Polohování na povrchu	
	2.2.10.19.1 Příklad kódu	
	2.2.10.20 Vypnuta kontrola flexibility	
	2.2.10.21 Kontrola flexibility na	
	2.2.10.21.1 Priklad kodu	
	2.3 Srovnávací tabulka chybových kôdů	
3	M cobot_ros	419
	3.1 Přehled	
	3.2 Instalace	
	3.2.1 Požadavky na životní prostředí	
	3.2.2 Instalace a požadavky na ROS	
	3.2.3 Kompilace	
	3.3 Rychlý start	
	3.3.1 M cobot_hw	
	421	
4	M cobot_ros2	423
	4.1 Přehled	
	4.2 M _ros2	
	4 2 1 Základní instalace prostředí	423
	4.2.2 Kompilace a sestavení	423
	4.3 Rychlý start	424
	4.3.1 Start	424
	4.3.2 Zobrazení zpětné vazby o stavu robotického ramene	424
	4.3.3 Objednávka vydání	425
	4.3.4 Upravit narametr	427
	4.4 Popis API	427
5	Brožura	437
-		
6	Certifikace kvalifikace	439
7	Sekundární vývoj	441
0	Ontologie a norměnové výlvrou	447
ð	Ontologie a rozinerove vykresy	443
9	3D modely	445

KAPITOLA

ONE

## KOLABORATIVNÍ ROBOT

## 1.1 Matice produktů



## 1.2 Rychlý start

### 1.2.1 Instalace a zapnutí robota

#### 1.2.1.1 Instalace robotického ramene

Při instalaci kolaborativního robota na montážní základnu použijte k utažení a upevnění robota na montážní základnu vyhovující počet šroubů (pevnost nejméně 8,8); doporučuje se použít dva vyhovující otvory pro kolíky a kolíky na montážní základně pro umístění robota, aby se zlepšila přesnost instalace robota a zabránilo se pohybu robota M om v důsledku kolizí. Pokud má robot vysoké požadavky na přesnost chodu, nezapomeňte přidat kolíky pro umístění robota.

Model kolaborativního robota	Šroub	Krouticí moment Šroubu	Specifika otvorů pro kolíky
M3	4 kusy M6	10Nm	5 mm
M5	4 kusy M8	20 Nm	8 mm
M10	4 kusy M8	25 Nm	8 mm
M16	4 kusy M8	25 Nm	8 mm
M20	6 kusů M10	45 Nm	8 mm

#### Tabulka 1.1-1 Standardní montážní díly robota

**Důležité:** Doporučujeme, aby montážní základna robota splňovala následující požadavky, aby byla zajištěna pevná a stabilní instalace robota:

(1) Držák robota musí být dostatečně pevný a mít dostatečnou nosnost. Měl by být schopen unést alespoň pětinásobek hmotnosti robota a alespoň desetinásobek točivého momentu v 1 ose.

(2) Povrch montážního sedla robota by měl být rovný, aby byl zajištěn těsný kontakt s kontaktní plochou robota;

(3) Montážní základna robota by měla být dostatečně pevná, pevně uchycená a neměla by s robotem rezonovat;

(4) Při současném pohybu robota a dalších dílů by měla být montážní základna izolována od ostatních pohyblivých dílů a neměla by být upevněna společně, aby se zabránilo rušivým vibracím během pohybu;

(5) Pokud je robot instalován na mobilní plošině nebo externí ose, mělo by být zrychlení mobilní plošiny nebo externí osy co nejmenší;

#### 1.2.1.2 Připojení ovládací skříňky

Tato řada oMobotů je napájena jednofázovým zdrojem střídavého proudu TN-S 220 V. Zařízení se dodává s 5metrovým napájecím kabelem. Tříkolíkový konec zástrčky se zasune do zásuvky AC 220V, která je k dispozici na místě, a robot je elektricky uzemněn. Všechny vnější přípojky řídicího systému manipulátoru jsou připojeny pomocí zástrček, které lze rychle instalovat. Panel zapojení kolaborativního robota je následující:



Obrázek 1.2-1 Panel zapojení ovládací skříňky

Rozhraní tlačítkové skříňky je ve výchozím nastavení řídicím portem učícího přívěsku a IP adresa je 192.168.58.2. K propojení rozhraní tlačítkové skříňky a počítače použijte síťový kabel. IP adresa počítače je nastavena na 192.168.58.10 nebo na stejný segment sítě. Otevřete prohlížeč Google a zadejte adresu 192.168.58.2. Můžete se dostat na stránku s přívěskem učení.

#### 1.2.1.3 Znát pole tlačítek a koncovou LED diodu

#### 1.2.1.3.1 Tlačítkový rámeček

Obrázek 1.3-1 Druhá verze tlačítkové skříňky

Tabulka 1.3-1 Popis tlačítek na panelu zapojení ovládací

-1-	× /	×.	1
SK	rı	n	KV

Název	Funkce
Spínač nouzového zastavení	Po stisknutí spínače nouzového zastavení přejde robot do stavu nouze. zastavit.
Spuštění/zastavení	Spuštění/zastavení spuštěného programu.
Ethernet	Připojte se k webovému osciloskopu.
Vypnout	Není povoleno.
Bod záznamu	Zaznamenejte výukový bod.
Režim výuky	Vstup/výstup se stavem výukového přívěsku.
Pracovní režim	Přepínač automatického/ručního režimu.
Režim přetahování	Vstup/výstup z režimu přetahování.



#### 1.2.1.3.2 Koncová LED dioda

Funkce	Barva LED
Když není navázána komunikace	Střídavě "Vypnuto", "Červená", "Zelená" a "Modrá".
Automatický režim	Modrá dlouhá světlá
Manuální režim	Zelená dlouhá světlá
Režim přetahování	Bílá azurová dlouhá světlá
Bod záznamu tlačítkového pole (pouze při použití tlačítkového pole)	Fialová dvakrát blikne
Spustit běh (pouze při použití tlačítkového pole)	Azurově modrá dvakrát zabliká
Zadejte stav neshodného tlačítkového pole (pouze v případě, že se jedná o us- v poli tlačítek)	Modrá dvakrát zabliká
Zastavení provozu (pouze při použití tlačítkového pole)	Červená dvakrát blikne
Hlášení chyb (pouze při použití tlačítkového pole)	Červená dlouhá světlá
Kalibrace nuly dokončena	Bílá azurová třikrát blikne
Povolit	Žlutá dvakrát blikne

Tabulka 1.3-2 Tabulka definic koncových LED diod

#### 1.2.1.4 Povolení zapnutí

Před zapnutím zkontrolujte, zda je uvolněno tlačítko nouzového zastavení tlačítkové skříňky, stiskněte červené spínací tlačítko ovládací skříňky pro zapnutí a kontrolka LED na konci bude po úspěšném zapnutí v trvalém zeleném stavu.

#### 1.2.2 Přihlášení k přístupu k webové aplikaci

#### 1.2.2.1 Přístup k rozhraní WebApp a přihlášení k němu

- 1. Zapněte řídicí jednotku a připojte síťový kabel k počítači;
- 2. Otevřete prohlížeč Chrome na počítači a přejděte na cílovou adresu URL 192.168.58.2;
- 3. Zadejte uživatelské jméno a heslo a kliknutím na tlačítko Přihlásit se

přihlaste do webové aplikace. Počáteční uživatelské jméno je admin a heslo je

123.

#### Obrázek 2.1-1 Přihlašovací rozhraní

#### 1.2.2.2 Jednoduché pochopení rozhraní aplikace WebApp

Po úspěšném přihlášení systém přejde do "úvodního rozhraní". Počáteční rozhraní ukazuje, že výukový přívěsek obsahuje především LOGO M INNOVATION a tlačítko pro návrat na počáteční stránku, lištu nabídek, tlačítko pro zvětšení lišty nabídek, oblast ovládání robota, oblast ovládání, stavovou oblast, oblast 3D simulace robota a oblast informací o poloze a IO, celkem osm oblastí. Počáteční rozhraní systému je znázorněno na obrázku níže:

#### 1.2.2.2.1 Kontrolní oblast





název: Tlačítko Start

účinek: Nahrajte a spusťte výukový program



název: Tlačítko Stop

účinek: Zastavení aktuálního výukového programu



#### Poznámka.

název: **Tlačítko pauzy/pokračování** účinek: Pozastavení a obnovení aktuálního výukového programu

#### 1.2.2.2.2 Stavový řádek



#### Poznámka:

název: stav robota

účinek: Zastaveno-stopBěží-běžíPauza-pauzaPřetáhnout-přetáhnout

# Toolcoord1

#### Poznámka:

název: Číslo souřadnicového systému nástroje

účinek: Zobrazení čísla souřadnicového systému nástroje aktuální aplikace



#### Poznámka:

Poznámka:

název: Rychlost běhu v procentech

účinek: Rychlost robota při jízdě v aktuálním režimu.



název: **Běží normálně** účinek: Aktuální robot běží normálně



Poznámka:

název: Error state

účinek: V aktuální operaci robota došlo k chybě



Poznámka:

#### název: automatický režim

účinek: Když je zapnuto globální nastavení rychlosti v manuálním a automatickém režimu a je zadána rychlost, globální rychlost se automaticky nastaví na zadanou rychlost.



#### Poznámka:

#### název: Teach mode

účinek: Nastavte globální nastavení rychlosti manuálního režimu a automatického režimu a určete rychlost.



#### Poznámka:

#### název: Drag state

účinek: Aktuální robot může přetáhnout



#### Poznámka:

název: **Drag state** účinek: Aktuální robot není tažný



Poznámka:

název: **Stav připojení** účinek: Robot připojen



**Poznámka:** název: **Not connected status** účinek: Robot není připojen



Poznámka:

```
název: Informace o účtu
```

účinek: Zobrazení uživatelského jména a oprávnění a odhlášení uživatele

#### 1.2.3 Nastavení parametrů robota

#### 1.2.3.1 Nastavení způsobu instalace

Výchozím režimem instalace robota je horizontální instalace. Při změně režimu instalace robota je třeba na této stránce včas nastavit aktuální režim instalace robota, aby byl zajištěn normální provoz robota.

Uživatel klikne na záložku "Pevná montáž" v oblasti 3D virtuálního zobrazení robota a zadá pevnou montáž robota. Na stránce nastavení režimu vyberte "Montáž napevno", "Montáž naklopením" nebo "Boční montáž" a kliknutím na tlačítko "Použít" dokončete nastavení režimu instalace robota.



#### Obrázek 3.1-1 Pevná montáž

S ohledem na flexibilnější a bohatší scénáře nasazení robotů poskytujeme funkci montáže M ee. Kliknutím na záložku "360-stupňová M ee montáž" v oblasti 3D virtuálního zobrazení robota vstoupí uživatelé na stránku nastavení režimu M ee montáže robota. Ručně nastavte úhly "náklonu základny" a "natočení základny" a 3D model podle toho zobrazí efekt montáže. Po úpravě kliknutím na tlačítko "Apply" (Použít) dokončete nastavení způsobu montáže robota.



Obrázek 3.1-2 Montáž v rozsahu 360 stupňů M ee

**Důležité:** Po dokončení instalace robota je nutné správně nastavit způsob instalace robota, jinak to ovlivní používání funkce tažení robota a funkce detekce kolizí.

#### 1.2.3.2 Nastavení koncového zatížení

Na panelu nabídek "Robot Settings" v části "Initialize" klikněte na "End Load", čímž vstoupíte na stránku End load.

Load weight setting				
Load weight	2.000	kg		
		Apply		
Load centroid	l coordinate se	etting		
X 0.000	Y 0.000	Z 150.000		
*Input range of mm	centroid coordina	ate - 1000~1000, uni		
Empty load		Apply		

Obrázek 3.2-1 Schéma nastavení zátěže

Uživatel může nastavit odpovídající parametry podle parametrů použitého nástroje. Hmotnost břemene je 0~5 kg a rozsah souřadnic středu hmotnosti je 0~1000, jak je znázorněno na obrázku 3.2-1.

**Důležité:** Po instalaci zátěže na konec robota je třeba správně nastavit hmotnost koncové zátěže a souřadnice středu hmotnosti, jinak to ovlivní funkci odporu robota a použití funkce detekce kolize.

#### 1.2.3.3 Nastavení souřadnic nástroje

V nabídce "Robot Settings" (Nastavení robota) v části "Initialize" (Inicializovat) klikněte na položku "Tool Coordinates" (Souřadnice nástroje), čímž vstoupíte na stránku Tool Coordinates Page (Souřadnice nástroje). Souřadnice nástroje mohou realizovat úpravu, vymazání a použití souřadnic nástroje. V rozevíracím seznamu souřadnicového systému nástroje je 15 čísel. Po výběru odpovídajícího souřadnicového systému (název souřadnicového systému lze přizpůsobit) se níže zobrazí odpovídající hodnota souřadnic, typ nástroje a pozice instalace (zobrazí se pouze u nástroje typu senzor), po výběru určitého souřadnicového systému klikněte na tlačítko "Použít" a aktuálně používaný souřadnicový systém nástroje se změní na vybrané souřadnice, jak je znázorněno na obrázku 3.3-1.

Kliknutím na tlačítko "Modify" obnovíte souřadnicový systém nástroje čísla podle výzvy. Metoda kalibrace nástroje se dělí na čtyřbodovou a šestibodovou metodu. Čtyřbodová metoda kalibruje pouze nástroj TCP, tedy polohu středového bodu nástroje. Její poloha je ve výchozím nastavení shodná s koncovou polohou. Šestibodová metoda přidává ke čtyřbodové metodě dva body., které se používají ke kalibraci polohy nástroje.

Coordinate name	system	toold	coord7	•
X 0.000	Y	0.000	Z	200.000
RX 0.000	RY	0.000	RZ	0.000
Tool Type:	0		0: too	il, 1: sensor
Installati	0		0: en	d, 1: external

Obrázek 3.3-1 Nastavení souřadnic nástroje

Tool Type: tool	
Modify Wizard	
Four point method	Six point method
	O Setpoint 1
	O Setpoint 2
2	O Setpoint 3
3	O Setpoint 4
	O Setpoint 5
	O Setpoint 6
	Calculate

Obrázek 3.3-2 Nastavení souřadnic nástroje

**Důležité:** 1. Po instalaci nástroje na konci je třeba kalibrovat a použít souřadnicový systém nástroje, jinak poloha a poloha středového bodu nástroje nebudou odpovídat očekávaným hodnotám, když robot provede příkaz k pohybu.

2.Souřadnicový systém nástroje obecně používá toolcoord1~toolcoord14 a toolcoord0 se používá k označení, že polohový střed nástroje TCP je ve středu koncové příruby. Při kalibraci souřadnicového systému nástroje je nejprve nutné použít souřadnicový systém nástroje toolcoord0 a poté zvolit další souřadnicové systémy nástroje pro kalibraci. Kalibrace a aplikace.

#### 1.2.4 Ruční výuka robotů

#### 1.2.4.1 Ruční výuka a záznam výukových bodů

Ruční výuka zahrnuje dvě metody, jednou je stisknutí a podržení koncového tlačítka pro přetažení a výuku a druhou je běh v operační oblasti. Po učení do cílové polohy lze učební bod uložit. Při ukládání výukového bodu je souřadnicovým systémem výukového bodu souřadnicový systém aktuální aplikace robota. Rychlost a zrychlení výukového bodu lze nastavit nad operační oblastí. Nastavená hodnota je procento standardní rychlosti robota. Pokud je nastavena na 100, jedná se o 100 % standardní rychlosti.



Obrázek 4.1-1 Ruční výuka

#### 1.2.4.2 Zobrazit informace o výukovém bodu

Kliknutím na "Správa výuky" zobrazíte všechny uložené informace o výukových bodech. V tomto rozhraní můžete importovat a exportovat soubory výukových bodů. Po výběru výukového bodu kliknutím na tlačítko "Delete" (Odstranit) informace o bodu odstraníte. Hodnoty výukových bodů x, y, z, rx, ry, rz a v lze upravit. Zadejte upravenou hodnotu, zaškrtněte modré políčko vlevo a kliknutím na horní modifikaci upravte informace o výukovém bodu. Kliknutím na tlačítko "Spustit běh" proveďte jednobodovou operaci místního výukového bodu a přesuňte robota do polohy tohoto bodu. Kromě toho mohou uživatelé vyhledávat výukové body podle názvu.

=			(	<del>9</del> 10									Stopped	toolcoord0	wobj0 e	exaxis0 10		1	2
Initialize ⊀	Teachin	g Managei	ment																
] Teaching 💙	Import	Export	Modity	Dekte															
	Name Sea	rch																	
regular callong	D	Name	x	۲	z	RX	RY	RZ	J1	J2	۵,	.14	J5	.6	TOOL	WOBJ	٧	Opr	erate
kaphical Hibgram	D	Ttart .	-547.837	29,059	123.975	-91,794	0.770	-92.430	-28.369	-72.041	121.795	-50.526	62.639	-44,613	1001000103	wolkpieceo	100	۲	• •
lanage Teaching	D	10111	-547.844	30.405	117 171	-76.793	0.771	-91.429	-28.511	-74.362	105.261	-14.765	62,000	-52,196	toolcoord3	workprece0.	100	۲	
Status <	D	tta/12	-517.841	38.408	.111.161	-70.794	0.771	-91.429	-30.962	-78.242	105.385	-3.892	61.545	-56,122	toolcoord3	workpiece0	100	۲	
Aunitary K	ъ	10013	-480.315	30.410	111.159	-70.794	0.772	-91.429	-35.660	-85.664	114.451	-4.140	\$6.329	-58.626	toolcoord3	wompiece0	100	۲	
Settings	D	1tail:	-455.679352	39 326639	113.092445	-81.678628	1.060899	-90 377373	-49,169	-91,011	133.024	-27.570	41.072	-56.042	toolcourd3	workpiece0	100	۲	
	D	twit	98 956 897	-312.020508	593.540405	-179 678940	-1.963225	0.1897.47	89 638	-92.426	119 538	-115.693	90.734	-43.580	tookoorca	workpiece0	100	۲	
	0	1wi/2	96.795090	-308.837006	593,592957	178.072754	-2.575468	93.845573	89.638	-92.426	119.538	-115.694	90.734	50.009	toorcoerca	0sosignow	100	۲	
	D	2tart	-546 226	01.892	287.717	-90.354	0.549	-92.634	-28.909	-89.149	95.347	-5.354	62.520	-45.103	toolcoord3	workpiece0	100	۲	•
	D	25412	-541 231	32 385	285 692	-74.494	0.550	-92.633	-28 182	-80.515	68.852	50 115	64.541	-52.563	toolcoorco	workpiece0	100	æ	•
	D.	7ta/3	-511.225	32.367	265 805	-69.452	0.552	-92 634	-30.797	-63.359	71.943	15.574	62 987	-55.964	tpolopord3	workpiece0	100		•
		These	461 200	19 385	265 810	55.455	0.554	80 634	-95-117	.44 213	89,490	17 394	55.425	80 135	tookoope	worknessed	100		
		Anna	117 100674	.174.456.517	ens 1300.10	179 904011	+ 490093	5.370040	00.800	101 524	+17.069	-105 141	80.204	19.509	testesee	-	100		
	0	theory.	69 944E30	202 200046	200 +97700	05 2966 92	0.02020.04	172 016639	91 519	05 190	70.007	78.440	80.024	12.597	Inclusion				
		Amp2	750 710510				2.020304	88 38 (375	02.320	-99,109	10.007	457.440	757724	-+2:007	Industrial	mungheter.	100		0
	10	Cirrus .	200.740019	-120.100702	357.041913	-2.011194	-2.0/3043	00.301278	09.120	-57.138	10.097	-107.190	21.235	~42.014	IU-HEDOFOT	mungleceu	100		0
Successful	U	zwod	202.285599	108.007591	487.753479	89.299753	0.030905	-89.201752	5.959	-114 894	124.066	-168.822	86.601	43.662	toolcoord3	wongrece0	100	۲	0
leaching point		zwig	252,215651	-108.012054	620.447693	89.274215	0.053055	-89.201569	2.930	-123.551	87/048	-143.143	86,500	-43.640	toolcoord3	womplece0	100	۲	0
- Cata -		2WU6	270.723328	-107.069771	516.709460	69.271538	-0.210132	-89.263046	2.930	-123 662	87.648	-163.159	86.649	-43.640	toolcoord3	workpiece0	100	۲	0

Obrázek 4.2-1 Rozhraní pro správu výuky

Důležité: Upravené hodnoty výukových bodů x, y, z, rx, ry, rz by neměly překročit pracovní rozsah robota.

#### 1.2.5 Rychlé programování robotů

#### 1.2.5.1 Úvod do jednoduchých pohybových instrukcí

Příkaz PTP: Kliknutím na ikonu "PTP" vstoupíte do rozhraní pro úpravu příkazů PTP.

Můžete zvolit bod, kterého má být dosaženo, a nastavením doby plynulého přechodu lze zajistit, aby pohyb M om tohoto bodu do dalšího bodu byl plynulý. Zda nastavit posun, můžete zvolit posun na základě základního souřadnicového systému a na základě souřadnic nástroje a vyskočí nastavení posunu x, y, z, rx, ry, rz, specifická dráha PTP je optimální dráha automaticky naplánovaná řídicí jednotkou pohybu, kliknutím na "Add" a "Apply" tento příkaz uložte.

<b>≣</b>		$(\mathbf{F})$	<u>ला</u>	P X	opped tooicoord0 wobj0 exaxis	0 10 1 🖌 🕅 ⊄ 🛞
@ Initializo <	🖬 🗅 🗄		Point Name: Tool coordinate system:	0731backup	Operation&Status 300" Pree Muturiting	faed Mounting
E Teaching ~	Al 🛛 👻	🕒 04444.kua	Workpiece coordinate	0	Move C	Robot Pose
Program Teachin	v ↔	1++ PTP(0803zj01,100,0,0)	J1	32.986		Joints J1:23.601 J2:-61.737 J3:50.330
Graphical Program	/ (L)	2↔ PTP(0803z)02,100,100,0)	J2	-96.861	'15"Z	J4 : -121,778 J5 : -67,575 J6 : -125,908
Manage reaching	Grito Wait	3→ PTP(0803zj04,100,100.0)	J3	54.350		K : 370.465 Y : 277.777 Z : 177.626 RX: 179.91 RV: 3.960 RZ: 120.349
394 Status 🤇	Pause Dofie	5→ PTP(0731backup.1001.0)	J4	-49.866	23.601	FT
🗙 Sattings	(1:1) Var	6⊷ ₽TP(0803z)06,100,0,0)	70	-86.659	61.737	Tool Fe 0.000 Py 0.000 Fz 0.000 Te 0.000 Ty 0.000 Tz 0.000
	Motion command		Commissioning speed	100 %	00.338	fuct_State:0
	PTTP LUN		At this point 🔹 Stop	O Smooth 0 ms	-121.778	Line-Num
	ARC Cede		Offset or not	<u>no</u>	-125.966	CtriBox
	88			Add		004 005 008 007 000 001 002 003
	Spral N-Spine		Added Commands:		Add	DIO OU DIO DIS DIA DI6 DI6 DI7

#### Obrázek 5.1-1 Příkazové rozhraní PTP

**Příkaz Lin**: Klepnutím na ikonu "Lin" vstoupíte do rozhraní pro úpravu příkazu Lin. Funkce tohoto příkazu je podobná příkazu "PTP", ale cesta bodu dosaženého tímto příkazem je přímka.



Obrázek 5.1-2 Příkazové rozhraní Lin

#### 1.2.5.2 Operace s programovými soubory

Upravte strom programů pomocí panelu nástrojů v dolní části stromu programů.



**Poznámka:** název: **Open** účinek: Otevřít soubor uživatelského programu



Pozn**á**mka:

název: New build

účinek: Výběr šablony pro vytvoření nového programového souboru


## Poznámka:

#### název: Import

efekt: Import souboru do složky uživatelského programu



## Poznámka:

#### název: Export

účinek: Export souborů uživatelských programů do místního bodu.



#### Poznámka:

název: Save

účinek: Uložit úpravy souboru



#### Poznámka:

název: Uložit jako

účinek: Přejmenujte soubor a uložte jej do složky uživatelského programu nebo šablony programu.



## Poznámka:

#### název: Copy

účinek: Duplikuje uzel a umožňuje jeho použití pro další operace (např.: vložení na jiné místo ve stromu programu).



## Poznámka:

#### název: Paste

účinek: Umožňuje vložit dříve vystřižené nebo zkopírované uzly.



#### Poznámka:

## název: řezat

účinek: Vložení uzlu do jiného místa ve stromu programu: Vyřízne uzel a umožní jeho použití pro jiné operace (např.: vložení na jiné místo ve stromu programu).



#### Poznámka:

#### název: Smazat

účinek: Odstraní uzel M ze stromu programu.



#### Poznámka:

#### název: Move up

účinek: Posuňte uzel nahoru.



## Poznámka:

název: **Přesun dolů** účinek: Posuňte uzel dolů.



## název: **Přepnutí režimu úprav**

Poznámka:

účinek: Režim stromu programů a režim editace lua se navzájem přepínají.

#### 1.2.5.3 Napsat a spustit program

Levá strana slouží především k přidávání programových příkazů. Kliknutím na ikonu nad každým klíčovým slovem vstoupíte do podrobného rozhraní. Existují dvě hlavní operace pro přidávání programových příkazů do souboru. Jedním způsobem je otevřít příslušný příkaz a kliknutím na tlačítko Použít přidat příkaz do programu., druhým způsobem je nejprve kliknout na tlačítko "Přidat", v tomto okamžiku se příkaz neuloží do souboru programu a je třeba znovu kliknout na tlačítko "Použít", aby se příkaz uložil do souboru. Druhý způsob se často vyskytuje v případě, že je vydáno více příkazů stejného typu. K tomuto typu příkazu přidáme tlačítko Přidat a zobrazíme obsah přidaného příkazu. Klepnutím na tlačítko Přidat přidáte příkaz a přidaný příkaz zobrazí všechny přidané příkazy., klepnutím na tlačítko Použít uložíte příkaz do otevřeného souboru vpravo.

Kliknutím na tlačítko Start program spustíte; kliknutím na tlačítko Stop program zastavíte; kliknutím na tlačítko Pauza/obnovení program pozastavíte/obnovíte; když program běží, je aktuálně prováděný uzel programu zvýrazněn šedě.

V ručním režimu kliknutím na první ikonu na pravé straně uzlu přimějete robota, aby instrukci provedl sám, a druhá ikona slouží k úpravě obsahu uzlu.



Obrázek 5.3-1 Stromové rozhraní programu

# 1.3 Manuální

## 1.3.1 Předmluva

Především vám velmi děkujeme, že jste si zakoupili výrobky naší společnosti z řady kolaborativních robotů M. Naše výrobky byly pečlivě navrženy a mnohokrát testovány, abychom zajistili, že budou ve všech ohledech vyhovovat vašim potřebám.

Pečlivě si přečtěte tuto uživatelskou příručku, abyste se ujistili, že můžete naše výrobky správně používat a získat co nejlepší zkušenosti. Pokud se během používání setkáte s jakýmikoliv problémy, podívejte se na část o vyloučení závady v návodu nebo se obraťte na našeho poprodejního pracovníka. Velmi vám děkujeme za vaši podporu a důvěru a těšíme se, že vám budeme poskytovat lepší služby a výrobky.

## 1.3.1.1 Co je nainstalováno v krabici

Při objednání kolaborativního robota řady M obdržíte krabici. Obsahuje:

- Jeden kolaborativní robot
- Jedno tlačítkové pole
- Jeden ovládací box (včetně jednoho kabelu ovládacího boxu)

## 1.3.1.2 Důležitý popis zabezpečení

Roboti jsou zařízením, které se týká osobní bezpečnosti, a proto je třeba po každé instalaci robota provést posouzení bezpečnosti. Je nutné dodržovat všechny bezpečnostní popisy uvedené v kapitole 1.

## 1.3.1.3 Jak používat tuto příručku

Tato příručka obsahuje návodné informace o programování instalace robota. Příručka obsahuje:

- Instalace hardwaru částInstalace strojů a elektrická instalace oMoboty
- Analýza softwaru teachorVýuka a programování robotů

Tato příručka je určena integrátorům robotů a integrátoři by měli absolvovat základní mechanické a elektrické školení a měli by být obeznámeni se základními koncepty programování.

## 1.3.1.4 Dodržujte příslušné normy

Standardní	Definice
2006/42/EC:2006	Směrnice o strojních zařízeních: Směrnice Evropského parlamentu a Rady 2006/42/ES
	Rady ze dne 17. května 2006 o strojních zařízeních a o změně směrnice 95/16/ES (přepracované znění)
2004/108/EC:2004	Směrnice EMC: Směrnice Evropského parlamentu a Rady 2004/108/ES o elektromagnetické kompatibilitě.
	Rady ze dne 15. prosince 2004 o sbližování právních předpisů členských států týkajících se elektromagnetické kompatibility a o zrušení směrnice 89/336/EHS
EN ISO 13850:2008	Bezpečnost strojního zařízení: Nouzové zastavení - zásady pro konstrukci
EN ISO 13849-1:2008	Bezpečnost strojního zařízení: Část 1: Obecně zásady navrhování
EN ISO 13849-2:2012	Bezpečnost strojního zařízení: Část 2: Validace
EN ISO 12100:2010	Bezpečnost strojního zařízení: Obecné zásady navrhování, posuzování rizik a opětovného riskování dukce
EN ISO 10218-1:2011	Průmyslové roboty: Obsah odpovídá normě ANSI/RIA R.15.06-2012, Část 1
ISO/TS 15066: 2016	Bezpečnostní požadavky na kolaborativní průmyslové roboty Roboti a robotická zařízení
	-Spolupracující roboti

## 1.3.2 Stručný úvod o robotech

## 1.3.2.1 Základní parametry

Tabulka 1.1-1 Základní parametry oMobotics

## 1.3.2.2 Pohyb

Prostor pro instalaci robotického ramene:

Instalace těla robota vyžaduje prostor o rozměrech 3 m × 3 m × 2 m (délka × šířka × výška), aby vyhovoval pohybu při maximálním rozpětí křídel robota; pokud uživatel sám zvýší koncové zatížení, ujistěte se, že v prostoru pro instalaci je mezera minimálně 500 mm.

**Poznámka:** Výškový prostor je ovlivněn výškou instalační základny, přičemž 2 m se vztahují ke vzdálenosti nad referenční rovinou instalace.

Prostor pro instalaci rozváděče:

1. Ovládací skříňka by měla být umístěna na snadno ovladatelném místě, aby nedošlo k zaplavení vodou, 0,6-1,5 m od země.

2. Skříň musí být daleko od zdroje tepla.

3. Na jedné straně linie těžkého zatížení ovládací skříně by neměla být žádná překážka do vzdálenosti 150 mm a zbývající strana nesmí být zakryta do vzdálenosti 100 mm, což je vhodné pro rozptyl a odvod tepla.

Obrázek 1.1-1 Rozsah pohybu kolaborativního robota modelu M 3 Obrázek 1.1-2 Rozsah pohybu kolaborativního

robota modelu M 5 Obrázek 1.1-3 Rozsah pohybu kolaborativního robota modelu M 10 Obrázek 1.1-4 Rozsah pohybu kolaborativního robota modelu M 16

Model Parameter	FR3	FR5	FR10	FR16	FR20		
Paytoad	3 kg	Skg	l0 kg	l6 kg	20 kg		
Dosáhnout	622mrn	922 mm	400 mm	1076 mm	1868 mm		
Rozsah		6 ro	6 rotačních otvorů				
Opakovatelno st	č.02mm	x0,02 mm	z0.O5mm	z0,03mm	zO.lmm		
	Ax1s : +175° , -175°:		AXIsI: +I7S°	, -175º:			
-	Ax1s2: -i-85° , -265°'		AXIs2: AxIs3	: +160° , -			
Pracovn í	Ax1s 3: +150° , -150°;		160°; Axls4:	+85°, -			
rozsah	Axls4: +AX1s 5: +175° ,		265°' Axls3:	+160°, -			
(soft limit)	-175°;		160°: +85°	, -265°;			
	Aks6i +175° ,-175"		AXIs 5: +175	°, -175°;			
			Osa 6: +175	° , -175°;			
	Osa 1: +17+° , -179º;		Osa 1: +179	9°, -179°;			
	Osa2: +89° , -269º:		Osa 2: -I-89				
Pracovní	Ax1s 3: +152° , -152°;		PXISZ: +I62	°, -162°;			
rozsan (tvrdý	Axls4: +89° , -y6g°;		Osy 4: +89°	, -269°;			
llmh)	AX1s S : +179° , -179";		Axls 5: +179'	, -179°;			
	Ax1s ó : +179° , -179º;		AXIs 6: +179°	°, -179°;			
Maximum speed	J80e/s	180°/s	Axisl,Z: I2D°/sOthe	erA s : 1B0°/s	/t4sl,2: 120°/s Ostatní osy: J8o°/s		
<b>TCP</b> rychlosւ	1m/s	1mis	1,5 m/s	lm/s	2m/s		
cbisslficaflon H	łluk	IP5	64 (volitelně IP66)				
Rc4ai:n mountlng			<65 dB				
		Jal	kákoli orientace				
Vstup a výstup	Napájen	ıí (24 V, 1,5 A), dig	itální IO, analogový	IO, komunikace 4	85		
komunikace	I/O. TCP/IP. Modbus_TCP/RTU. Profrinet						
Provozní teplota			0-45°C				
Provozní vlhkost		90%F	RH (nekondenzující)				
Hmotnost	Prů	opravy -15	ikg	=20,6 kg			
Skladován í vybavení	měr ná	porucny					

=40 kg =40 kg =65kg -25 60°C (bez mrazu) 2h  $\Gamma^{\prime}$ 



Ĵ.

 ${\mathbb P}^{*}$ 



Ĵ,



114

586

1

106



J.

 ${\mathbb P}^{n}$ 



Ĵ,



## Obrázek 1.1-5 Rozsah pohybu kolaborativního robota modelu M 20

## 1.3.2.3 Souřadnicový systém robota



Obrázek 1.1-6 Souřadnicový systém parametrů robota DH Obrázek 1.1-7 Souřadnicový systém M ench endMame

## 1.3.2.4 Parametry robota Denavit-Hartenberg

Denavit-Hartenbergovy parametry se používají k výpočtu kinematiky a dynamiky kolaborativních robotů řady FR. Obrázek 1.1-8 Kolaborativní roboty řady M Denavit-Hartenbergovy parametry Denavit-Hartenbergovy parametry kolaborativních robotů řady FR jsou uvedeny níže.

Tabulka 1.1-2 M 3 model spolupráce robotů Denavit-Hartenberg parametr

Kinematika	theta[ra	a[m]	d[m]	al- pha[rad]	Dynamics	Hmotnos [kg]	t Hmotnostní střed[m]
Joint1	0	0	140	/2	odkaz1	2.24	[-0.05, -15.92, 2.26]
Joint2	0	-280	0	0	odkaz2	4.94	[139.49, 0, 99.54]
Joint3	0	-240	0	0	odkaz3	2.29	[58.99, 0.08, 12.99]
Joint4	0	0	102	/2	odkaz4	1.56	[0.05, -2.33, 14.67]
Joint5	0	0	102	-/2	odkaz5	1.56	[-0.05, 2.33, 14.67]
Joint6	0	0	100	0	odkaz6	0.36	[-0.55, -1.11, -20.05]

Tabulka 1.1-3 M 5 model spolupráce robotů Denavit-Hartenberg parametr

J.





Kinematika	theta[ra	a[m]	d[m]	al-	Dynamics	Hmotnost Hmotnostní střed[m]	
				pha[rad]		[kg]	
Joint1	0	0	152	/2	odkaz1	4.64	[-0.19, -18.28, 2.26]
Joint2	0	-425	0	0	odkaz2	10.08	[212.47, 0, 121.2]
Joint3	0	-395	0	0	odkaz3	2.71	[122.62, 0.17, 12.59]
Joint4	0	0	102	/2	odkaz4	1.56	[0.05, -2.33, 14.68]
Joint5	0	0	102	-/2	odkaz5	1.56	[-0.05, 2.33, 14.68]
Joint6	0	0	100	0	odkaz6	0.36	[0.93, 0.81, -20.05]

Tabulka 1.1-4 M 10 model spolupráce robotů Denavit-Hartenberg parametr

Kinematika	theta[ra	a[m]	d[m]	al-	Dynamics	Hmotnos	t Hmotnostní střed[m]
				pha[rad]		[kg]	
Joint1	0	0	180	/2	odkaz1	11.97	[-0.10, -26.12, 4.04]
Joint2	0	-700	0	0	odkaz2	19.59	[480.27, 0.01, 164.68]
Joint3	0	-586	0	0	odkaz3	3.7	[211.22, 0.11, 54.21]
Joint4	0	0	159	/2	odkaz4	1.69	[0.12, -3, 12.18]
Joint5	0	0	114	-/2	odkaz5	1.69	[-0.12, 3, 12.18]
Joint6	0	0	106	0	odkaz6	0.35	[1.24, 0.85, -20.34]

Tabulka 1.1-5 M 16 model spolupráce robotů Denavit-Hartenberg parametr

Kinematika	theta[ra	a[m]	d[m]	al-	Dynamics	Hmotnost Hmotnostní střed[m]	
				pha[rad]		[kg]	
Joint1	0	0	180	/2	odkaz1	11.97	[-0.10, -26.12, 4.04]
Joint2	0	-520	0	0	odkaz2	18.18	[364.4, 0.01, 163.09]
Joint3	0	-400	0	0	odkaz3	3.22	[135.03, 0.12, 55.58]
Joint4	0	0	159	/2	odkaz4	1.69	[0.12, -3, 12.18]
Joint5	0	0	114	-/2	odkaz5	1.69	[-0.12, 3, 12.18]
Joint6	0	0	106	0	odkaz6	0.35	[1.24, 0.85, -20.34]

Tabulka 1.1-6 M 20 model spolupráce robotů Denavit-Hartenberg parametr

Kinematika	theta [rad]	a [m]	d [m]	alfa [rad]	Dynamics	Hmo tnost [kg]	Hmotnostní střed [m]
Joint1	0	0	215	/2	odkaz1	20.79	[-0.19, -36.57, 5.68]
Joint2	0	-1000	0	0	odkaz2	42.84	[605.25, 0.06, 202.94]
Joint3	0	-716	0	0	odkaz3	9.88	[262.84, 0.22, 43.08]
Joint4	0	0	166	/2	odkaz4	4.64	[0.23, -2.28, 18.42]
Joint5	0	0	138	-/2	odkaz5	4.64	[-0.23, 2.28, 18.42]
Joint6	0	0	120	0	odkaz6	0.6	[-2.11, -1.96, -20.38]

## 1.3.2.5 Stažení parametrů DH

Transformace MRobots DH

## 1.3.3 Instalace

## 1.3.3.1 Pokyny pro zabezpečení

## 1.3.3.1.1 Stručný úvod

V této příručce budou použita následující upozornění. Úkolem těchto varování je zajistit bezpečnost osob a zařízení. Při čtení této příručky musíte dodržovat a provádět všechny montážní pokyny a pokyny uvedené v dalších kapitolách této příručky. Je to velmi důležité. Zvláštní pozornost věnujte textu týkajícímu se výstražných značek.

#### Důležité:

- Pokud dojde k poškození, změně nebo úpravě robota (těla robota, ovládací skříňky, osciloskopu nebo tlačítkové skříňky) z an- tropogenních příčin, společnost M INNOVATION odmítá nést veškerou odpovědnost.
- Společnost MINNOVATION neodpovídá za škody způsobené robotem nebo jiným zařízením v důsledku chyb napsaných zákazníky nebo jiným zařízením.

## 1.3.3.1.2 Bezpečnost personálu

Při provozu robotického systému musíme nejprve zajistit bezpečnost obsluhy. Obecná bezpečnostní opatření jsou uvedena níže. Přijměte odpovídající opatření k zajištění bezpečnosti obsluhy.

- 1. Všichni operátoři, kteří používají robotický systém, by měli být proškoleni v rámci školení, které sponzoruje společnost M IN- NOVATION (Suzhou) ROBOTIC SYSTEM COMPANY. Uživatelé musí zajistit, aby plně pochopili bezpečné a standardizované provozní postupy a měli kvalifikaci pro robotické operace. Podrobnosti o školení si vyžádejte u naší společnosti, poštovní schránka je jiling@Mtech.M.
- 2. Všichni operátoři, kteří používají robotický systém, nenosí volné oblečení ani šperky. Při obsluze robota dbejte na to, aby byl paprsek dlouhých vlasů za hlavou.
- 3. I když se během provozu zařízení zdá, že se robot zastavil, může to být proto, že robot čeká na spuštění signálu a je v pohybu. I v takovém stavu by měl být robot považován za robota v činnosti.
- 4. Na podlaze by měly být nakresleny čáry označující rozsah robota, aby obsluha pochopila, že akční rádius robota zahrnuje i držení nástrojů (robotika, nářadí atd.).
- 5. Zajistěte bezpečnostní opatření (například zábradlí, lana nebo ochranné zástěny) v blízkosti prostoru pro práci s robotem, abyste ochránili obsluhu a okolní osoby. Podle potřeby by měly být zřízeny zámky, aby se osoby odpovědné za provoz nemohly dostat k napájení robota.
- 6. Při používání ovládacího panelu a osciloskopu, protože v rukavicích může dojít k provozní chybě, musíte pracovat po sundání rukavic.
- 7. V případě nouzových situací a abnormalit, kdy jsou lidé robotem sevřeni nebo zabaleni dovnitř, tlačí nebo táhnou robotické rameno silou (nejméně 700 N), aby si vynutili pohyby kloubů. S ovladačem Power-Mee je rameno ručního mobilního robota omezeno na nouzové situace a může dojít k poškození kloubu.

## 1.3.3.1.3 Nastavení zabezpečení

V nabídce "Safety Settings" (Nastavení bezpečnosti) v "Auxiliary Application" (Pomocná aplikace) klikněte na "Safety stop mode" (Režim bezpečnostního zastavení), čímž vstoupíte do rozhraní funkce nastavení bezpečného zastavení.

Povolte režim bezpečnostního zastavení. Když je robot hlášen nebo varován, automaticky se zapne a hraje roli bezpečné ochrany.

Safe stop	mode	
Whether		
enable	Not effective	

Obrázek 2.1-1 Nastavení bezpečnostního dorazu

• Snížení rychlosti: Po aktivaci tohoto režimu se omezí rychlost robotického ramene v prostoru kloubu. Hodnota v příslušném textovém poli je mezní hodnota rychlosti kloubů každého kloubu, rozsah nastavení 1, 2 a 3 kloubů je 15 ~ 150 °/s, 4, 5 a 6 kloubů je 15 ~ 180 °/s; omezení rychlosti pohybu robotického ramene v prostoru Descartes je hodnota omezení rychlosti TCP a nastavený rozsah je 0 ~ 80 mm/s.

#### 1.3.3.1.4 Rozpoznání nebezpečí

Posouzení rizik by mělo zohlednit všechny možné kontakty mezi obsluhou a roboty při běžném používání a předvídatelná nedorozumění. Krk, obličej a hlava obsluhy by neměly být vystaveny dotyku. Používání robotů bez použití periferních bezpečnostních ochranných zařízení vyžaduje první posouzení rizik, aby se určilo, zda příslušná nebezpečí budou představovat nepřijatelné riziko, jako např.

- Může hrozit nebezpečí použití pohonu s ostrým koncem nebo konektoru nástroje;
- Může hrozit nebezpečí léčby toxickými nebo jinými škodlivými látkami;
- Hrozí nebezpečí sevření prstu obsluhy základnou nebo kloubem robota;
- Nebezpečí srážky s roboty;
- Nebezpečí, že oMoboty nebo nástroje připojené ke konci nebudou upevněny na místě;
- Riziko nárazu mezi účinným zatížením a silným povrchem robota.

Integrace musí změřit taková nebezpečí a související úrovně rizik prostřednictvím hodnocení rizik a stanovit a zavést odpovídající opatření ke snížení rizika na přijatelnou úroveň. Upozorňujeme, že u konkrétních robotů se mohou vyskytovat další významná nebezpečí.

Kombinací inherentních bezpečnostních konstrukčních opatření používaných roboty M s bezpečnostními specifikacemi nebo hodnocením rizik prováděným integrovanými a koncovými uživateli se rizika spojená se společnými operacemi robotů M sníží na přiměřenou a proveditelnou úroveň. Prostřednictvím tohoto dokumentu lze integrovaným a koncovým uživatelům před instalací sdělit veškerá zbývající rizika robota. Pokud integrátor na základě posouzení rizik zjistí, že existuje konkrétní aplikace, která může mít nepřijatelná rizika, která mohou představovat nepřijatelná rizika, musí integrátoři odpovídajícím způsobem snížit rizika

musí být přijata opatření k odstranění nebo minimalizaci těchto nebezpečí, dokud se riziko nesníží na přijatelnou úroveň. Před přijetím příslušných opatření ke snížení rizika (je-li to nutné) není bezpečné jej používat.

Pokud je robot instalován v nesynergické instalaci (například při použití nebezpečného nástroje), může z posouzení rizik vyplývat, že integrovaný poskytovatel musí připojit další bezpečnostní zařízení (například bezpečnostní spouštěcí zařízení), aby zajistil bezpečnost personálu a zařízení při programování.

## 1.3.3.1.5 Informace na výrobním štítku

Obrázek 2.1-2 Model kolaborativního robota M 3 Obrázek 2.1-3 Model kolaborativního robota M 5 Obrázek 2.1-4 Model kolaborativního robota M 10 Obrázek 2.1-5 Model kolaborativního robota M 16 Obrázek 2.1-6 Model kolaborativního robota M 20

## 1.3.3.1.6 Účinnost a odpovědnost

Informace v této příručce neobsahují kompletní aplikaci robota z hlediska konstrukce, instalace a provozu, ani všechna periferní zařízení, která mohou ovlivnit bezpečnost tohoto kompletního systému. Konstrukce a instalace tohoto kompletního systému musí splňovat bezpečnostní požadavky stanovené normami a specifikacemi dané země instalace.

Integrovaný integrátor FAIR INNOVATION odpovídá za to, že dodržuje zákony a předpisy příslušných zemí a že v kompletní robotické aplikaci nehrozí žádné větší nebezpečí. To zahrnuje mimo jiné následující:

• Proveďte posouzení rizik celého robotického systému

- Připojení dalších strojních zařízení a doplňkového bezpečnostního vybavení definovaného na základě posouzení a definice rizik.
- Zavedení vhodných nastavení zabezpečení v softwaru
- Ujistěte se, že uživatelé nebudou měnit žádná bezpečnostní opatření.
- Ověřte si, že návrh a instalace celého robotického systému jsou správné.
- Jasný návod k použití
- Označte příslušné značky a kontaktní údaje integrátorů na robotu.
- Shromážděte všechny dokumenty v technickém souboru, včetně této příručky.

## 1.3.3.1.7 Omezená odpovědnost

Jakékoli bezpečnostní informace obsažené v této příručce nelze považovat za obecnou záruku bezpečnosti robota. I když dodržíte všechny bezpečnostní popisy, může dojít k poškození personálu nebo zařízení.

## 1.3.3.1.8 Výstražné značky v této příručce

Následující příznak definuje vysvětlení ustanovení o stupni nebezpečí obsažených v této příručce. U výrobku jsou rovněž použity stejné výstražné značky.

#### Důležité:



Nebezpečí: Označuje spotřebu energie, která může způsobit nebezpečí. Pokud se jí nevyhnete, může vést k úmrtí nebo vážnému poškození.

Důležité:

Nebezpečí úrazu elektrickým proudem: Jedná se o blížící se situaci, kdy hrozí nebezpečí úrazu elektrickým proudem.





## 1.3.3.1.9 Hodnocení před použitím

Po prvním použití robota nebo jeho úpravě je výchozí rychlost robota nižší než 250 mm/s. Nepřihlašujte se ke správci, abyste upravili rychlost pro vstup do vysokorychlostního režimu. Poté je třeba provést následující test. Potvrdí se, že všechny bezpečnostní vstupy a výstupy jsou správné a připojení je správné. Otestujte, zda jsou všechny připojené bezpečnostní vstupy a výstupy (včetně více strojů nebo sdílených zařízení robotů) v pořádku. Je tedy nutné:

- Otestujte, zda tlačítko nouzového zastavení a vstup dokáží robota zastavit a spustit brzdu.
- Otestujte, zda ochranný vstup může zastavit pohyb robota. Pokud je ochranný reset nakonfigurován, zkontrolujte, zda je před obnovením potřeba aktivace.
- Režim testovacího provozu může přepínat provozní režim, viz ikona v pravém horním rohu uživatelského rozhraní.

- Otestujte, zda je nutné stisknout ovládací zařízení 3. rychlostního stupně, aby se aktivovalo v manuálním režimu, a robot je pod kontrolou zpomalení (verze softwaru robota V3.0 tuto funkci nepodporuje).
- Otestujte, zda výstup nouzového zastavení systému dokáže uvést celý systém do bezpečného stavu.

## 1.3.3.1.10 Nouzové zastavení

Tlačítko nouzového zastavení je typu 0 stop. Stisknutím tlačítka nouzového zastavení okamžitě zastavíte všechny pohyby robota.

V následující tabulce je uvedena vzdálenost a doba zastavení zastávky typu 0. Tyto výsledky měření odpovídají následující konfiguraci robota:

- Vysunutí: 100 % (robotické rameno je plně roztažené)
- Rychlost: 100 % (obecná rychlost robota je nastavena na 100 %, pohybuje se kloubovou rychlostí 180 °/s)
- Efektivní zatížení: Maximální účinné zatížení

Kloub 1, kloub 6 testovací úrovně robota, rotující hřídel je kolmá k zemi. Kloub 2, kloub 3, kloub 4, kloub 5 testovacích robotů sleduje vertikální trajektorii, rotující hřídel je rovnoběžná se zemí a zastaví se, když se robot pohybuje dolů.

	Kloub 1	Joint 2	Společný 3	Společný 4	Společný 5	Společný 6
M3	0.47	0.60	0.56	0.29	0.10	0.06
M5	0.51	0.63	0.60	0.33	0.16	0.10
M10	0.64	0.70	0.69	0.42	0.25	0.13
M16	0.60	0.67	0.65	0.39	0.22	0.12
M20	0.69	0.75	0.80	0.48	0.31	0.22

Tabulka 2.1-1 Zastavovací vzdálenost kategorie 0 (rad)

#### Tabulka 2.1-2 Doba zastavení kategorie 0 (ms)

	Kloub 1	Joint 2	Společný 3	Společný 4	Společný 5	Společný 6
M3	400	470	450	280	120	90
M5	420	500	480	310	150	120
M10	460	540	510	330	170	140
M16	440	530	490	320	160	130
M20	540	600	700	400	260	170

Po nouzovém zastavení vypněte napájení, otočte tlačítkem nouzového zastavení a zapněte napájení, aby se robot znovu spustil.

Současně je v tabulce níže uvedena doba zastavení a vzdálenost zastavení bezpečnostního dorazu robota a měkkého koncového dorazu. Tyto výsledky měření odpovídají následující konfiguraci robota:

- Vysunutí: 100 % (robotické rameno je plně roztažené)
- Rychlost: 100 % (obecná rychlost robota je nastavena na 100 %, pohybuje se kloubovou rychlostí 180 °/s)
- Efektivní zatížení: Maximální účinné zatížení

Kloub 1, kloub 6 testovací úrovně robota, rotující hřídel je kolmá k zemi. Kloub 2, kloub 3, kloub 4, kloub 5 testovacích robotů sleduje vertikální trajektorii, rotující hřídel je rovnoběžná se zemí a zastaví se, když se robot pohybuje dolů.

Tabulka 2.1-3 Bezpečnostní brzdná dráha (rad)

	Kloub 1	Joint 2	Společný 3	Společný 4	Společný 5	Společný 6
M3	0.49	0.63	0.58	0.32	0.12	0.09
M5	0.54	0.65	0.63	0.35	0.19	0.12
M10	0.66	0.73	0.71	0.45	0.27	0.14
M16	0.63	0.69	0.68	0.41	0.25	0.14
M20	0.71	0.78	0.82	0.51	0.33	0.25

Tabulka 2.1-4 Doba bezpečnostního zastavení (ms)

	Kloub 1	Joint 2	Společný 3	Společný 4	Společný 5	Společný 6
M3	410	490	410	300	130	110
M5	450	520	510	330	180	140
M10	480	570	530	360	190	170
M16	470	550	520	340	190	150
M20	560	630	720	430	280	200

Tabulka 2.1-5 Vzdálenost měkkého mezního dorazu (rad)

	Kloub 1	Joint 2	Společný 3	Společný 4	Společný 5	Společný 6
M3	0.52	0.65	0.61	0.34	0.15	0.11
M5	0.56	0.68	0.65	0.38	0.21	0.15
M10	0.69	0.75	0.74	0.47	0.30	0.18
M16	0.65	0.72	0.70	0.44	0.27	0.17
M20	0.74	0.80	0.85	0.53	0.36	0.27

Tabulka 2.1-6 Doba zastavení měkkého limitu (ms)

	Kloub 1	Joint 2	Společný 3	Společný 4	Společný 5	Společný 6
M3	430	500	430	310	150	120
M5	460	540	520	350	190	160
M10	500	580	550	370	210	180
M16	480	570	530	360	200	170
M20	580	640	740	440	300	210

**Důležité:** Podle IEC 60204-1 a ISO 13850 není zařízení nouzového zastavení bezpečným ochranným zařízením. Jsou to doplňková ochranná opatření a nemusí zabránit poškození.

## 1.3.3.1.11 Hnutí Power-Mee

Pokud musíte kloubem robota pohybovat, ale nemůžete ho napájet nebo se vyskytnou jiné mimořádné situace, obraťte se na prodejce robota. V případě potřeby můžete použít násilné prostředky k donucení mobilních robotů k záchraně uvězněných osob.

## 1.3.3.2 Přeprava zařízení

## 1.3.3.2.1 Doprava

Robot a ovládací skříňky byly kalibrovány jako kompletní zařízení. Neoddělujte je od sebe, to by vyžadovalo opětovnou kalibraci.

Robota můžete přepravovat pouze v originálním obalu. Pokud budete chtít robota v budoucnu přemístit, uložte obalový materiál na suché místo.

Když se robot pohybuje M om obalu do prostoru instalace, obě ramena robota jsou držena současně. Držte robota tak dlouho, dokud nebudou všechny instalační šrouby uložení robota pevně utaženy.

## 1.3.3.2.2 Přenášejte

Podle různých modelů je celková kvalita (včetně obalu) 15-80 kg v závislosti na modelu. Při přepravě nebo přenášení kolaborativního robota lidskou silou je třeba, aby jej pomáhalo zvedat více lidí, nedoporučujeme manipulaci jednou osobou, během přepravy musí být stabilní. Zabraňte naklonění nebo sklouznutí zařízení.

## Varování:

- Pokud k manipulaci používáte profesionální vybavení, nezapomeňte při přepravě nebo přenášení kolaborativního robota použít jeřáb nebo vysokozdvižný vozík, jinak může dojít k poškození personálu nebo jiným nehodám;
- Pokud používáte ruční manipulaci, dbejte na osobní bezpečnost při manipulaci;
- Kolaborativní robot obsahuje přesné součásti, které by se měly vyhnout silným vibracím nebo otřesům během přepravy nebo transportu, jinak by mohlo dojít ke snížení výkonu zařízení.

## 1.3.3.2.3 Úložiště

Kolaborativní robot by měl být skladován při teplotě -25 ~ 60 °C a není zde prostředí M ost-Mee.

## 1.3.3.3 Údržba a zpracování šrotu

## 1.3.3.3.1 Údržba likvidace

Zkontrolujte nouzové zastavení a ochranné zastavení po dobu 1 měsíce. Zjistěte, zda je bezpečnostní funkce účinná.

Zapojení nouzového zastavení a ochranného zastavení naleznete v kapitole o zapojení.

## 1.3.3.3.2 Robotická likvidace odpadu

Mroboty je třeba likvidovat v souladu s platnými vnitrostátními zákony a předpisy a vnitrostátními normami. Pro podrobnosti se můžete obrátit na výrobce.

## 1.3.3.4 Specifikace instalace

## 1.3.3.4.1 Instalace ramene robota

**Důležité:** Doporučená základna pro instalaci robota splňuje následující požadavky, aby byla zajištěna bezpečná a stabilní instalace robota:

(1) Montážní základna robota musí být dostatečně pevná a mít dostatečnou nosnost, která by měla unést alespoň pětinásobek hmotnosti robota a alespoň desetinásobek točivého momentu v 1 ose.

(2) Povrch montážní základny robota by měl být rovný, aby byl zajištěn těsný kontakt s kontaktní plochou robota.

(3) Montážní základna robota by měla mít dostatečnou tuhost, měla by být pevně uchycena a neměla by s robotem rezonovat.

(4) Při současném pohybu robota a dalších součástí by měla být montážní základna oddělena od ostatních pohybujících se součástí a neměla by být upevněna společně, aby nedocházelo k rušení vibracemi během pohybu.

(5) Pokud je robot instalován na mobilní plošině nebo externí ose, mělo by být zrychlení mobilní plošiny nebo externí osy co nejmenší.

Varování: Vyvarujte se následujících způsobů instalace:

(I)Vyvarujte se připevnění robota k jiným pohyblivým zařízením.



Obrázek 2.4-1 Vyhněte se instalaci na jiné sportovní vybavení

Zkontrolujte, zda je rameno robota správně a bezpečně nainstalováno. Nestabilní instalace způsobí nehody.

**Poznámka:** Přesné podstavce můžete zakoupit jako přílohy. Obrázek 2.4-21.5-51.5-81.5-11 ukazuje polohu prodejního otvoru a umístění šroubu.

## 1.3.3.4.1.1 Požadavky na instalaci robota M 3

Při instalaci robota na montážní základnu použijte k upevnění robota na montážní základnu čtyři šrouby M6 o síle nejméně 8,8. Šrouby musí být utaženy krouticím momentem nejméně 10 Nm. doporučujeme použít na montážní základně dva 5mm otvory pro kolíky, které jsou přizpůsobeny kolíkům pro polohování robota, aby se zlepšila přesnost instalace robota a zabránilo se pohybu robota v důsledku kolizí a jiných faktorů. pokud má robot vysoké požadavky na provozní přesnost, nezapomeňte přidat kolíky pro polohování robota.



Obrázek 2.4-2 Velikost instalace kolaborativního robota modelu M 3

Důležité: Podle různých scénářů použití doporučujeme několik následujících základen pro instalaci robotů.

(I)V situacích, kdy rychlost pohybu není příliš vysoká, rychlost pohybu není příliš velká, požadavky na přesnost jsou průměrné a není vhodné robota upevnit na zem, je doporučená základna pro instalaci robota následující.



Obrázek 2.4-3 Základna pro montáž kolaborativního robota s nízkými nároky modelu M 3

(II) V situacích, kdy je rychlost pohybu vysoká, rychlost pohybu je vysoká a požadavky na přesnost jsou vysoké, se doporučuje instalovat robota na následující základnu a upevnit jej na pevnou podložku.



Obrázek 2.4-4 M 3 Model kolaborativního robota pro vysoké nároky Montážní základna

## 1.3.3.4.1.2 Požadavky na instalaci robota M 5

Při instalaci robota na montážní základnu použijte k upevnění robota na montážní základnu čtyři šrouby M8 s pevností nejméně 8,8. Šrouby musí být utaženy krouticím momentem ne menším než 20 Nm. doporučujeme použít dva na montážní základně 8mm otvory pro čepy sladěné s kolíky pro polohování robota, aby se zlepšila přesnost instalace robota a zabránilo se pohybu robota v důsledku kolizí a jiných faktorů. pokud má robot vysoké požadavky na provozní přesnost, nezapomeňte přidat kolíky pro polohování robota.



Obrázek 2.4-5 Instalační velikost kolaborativního robota modelu M 5

Důležité: Podle různých scénářů použití doporučujeme několik následujících základen pro instalaci robotů.

(I)V situacích, kdy rychlost pohybu není příliš vysoká, rychlost pohybu není příliš velká, požadavky na přesnost jsou průměrné a není vhodné robota upevnit na zem, je doporučená základna pro instalaci robota následující.



Obrázek 2.4-6 M 5 Model kolaborativního robota pro vysoké nároky Montážní základna

(II) V situacích, kdy je rychlost pohybu vysoká, rychlost pohybu je vysoká a požadavky na přesnost jsou vysoké, se doporučuje instalovat robota na následující základnu a upevnit jej na pevnou podložku.



Obrázek 2.4-7 Základna pro montáž kolaborativního robota s nízkými požadavky modelu M 5

#### 1.3.3.4.1.3 Požadavky na instalaci robota M 10&M16

Při instalaci robota na montážní základnu použijte k upevnění robota na montážní základnu čtyři šrouby M8 s pevností nejméně 8,8. Šrouby musí být utaženy krouticím momentem nejméně 25 Nm. doporučujeme použít dva na montážní základně 8mm otvory pro kolíky, které jsou přizpůsobeny kolíkům pro polohování robota, aby se zlepšila přesnost instalace robota a zabránilo se pohybu robota v důsledku kolizí a jiných faktorů. pokud má robot vysoké požadavky na provozní přesnost, nezapomeňte přidat kolíky pro polohování robota.



Obrázek 2.4-8 Instalační velikost kolaborativního robota modelu M 10&M16

Důležité: Podle různých scénářů použití doporučujeme několik následujících základen pro instalaci robotů.

(I)V situacích, kdy rychlost pohybu není příliš vysoká, rychlost pohybu není příliš velká, požadavky na přesnost jsou průměrné a není vhodné robota upevnit na zem, je doporučená základna pro instalaci robota následující.



Obrázek 2.4-9 Základna pro montáž kolaborativního robota s nízkými nároky modelu M 10&M16

(II) V situacích, kdy je rychlost pohybu vysoká, rychlost pohybu je vysoká a požadavky na přesnost jsou vysoké, se doporučuje instalovat robota na následující základnu a upevnit jej na pevnou podložku.



Obrázek 2.4-10 M 10&M16 Model Collaborative Robot High Demand Montážní základna

#### 1.3.3.4.1.4 Požadavky na instalaci robota M 20

Při instalaci robota na montážní základnu použijte k upevnění robota na montážní základnu šest šroubů M10 o síle nejméně 8,8. Šrouby musí být utaženy krouticím momentem nejméně 45 Nm. doporučujeme použít dva na montážní základně 8mm otvory pro kolíky sladěné s kolíky pro polohování robota, aby se zlepšila přesnost instalace robota a zabránilo se pohybu robota v důsledku kolizí a jiných faktorů. pokud má robot vysoké požadavky na provozní přesnost, nezapomeňte přidat kolíky pro polohování robota.

Obrázek 2.4-11 Instalační velikost kolaborativního robota modelu M 20

Důležité: Vzhledem k velké vlastní hmotnosti a setrvačnosti robota M 20 se doporučuje, aby byl robot přímo připevněn na konstrukci.



1

k použití na zemi. Doporučený základ je následující.





#### 1.3.3.4.2 Instalace koncovky nástroje

V robotickém nástroji jsou čtyři otvory se závitem M6, které lze použít k připojení nástroje k robotu. Šroub M6 musí být utažen krouticím momentem 8 nm a jeho pevnostní stupeň není nižší než 8,8. Pro přesné znovuzískání nářadí použijte hřebíky ve vyhrazených prodejních otvorech ø6.

> Obrázek 2.4-13 Výkres koncové příruby modelu robota M 3/M5/M10/M16 Obrázek 2.4-14 Výkres koncové příruby modelu robota M 20

#### Důležité:

- Ujistěte se, že je nářadí správně a bezpečně nainstalováno.
- Zajistěte bezpečnostní architekturu nářadí a žádné části dílů se nedostanou do nebezpečí.
- Instalace šroubů M6 o délce větší než 8 mm na přírubu robota může zničit příruby nástroje a způsobit poškození, které nelze opravit, což může vést k výměně nástroje.

E.



\_

г



1
# 1.3.3.4.3 Instalační prostředí

Při instalaci a používání kolaborativního robota dbejte na splnění následujících požadavků:

- Teplota prostředí 0-45 °C
- Vlhkost 20-80RH není vystavena
- Bez mechanických nárazů a otřesů
- Nadmořská výška vyžaduje méně než 2000 m
- Žádné korozivní plyny, žádné kapaliny, žádné výbušné plyny, žádné znečištění olejem, žádná solná mlha, žádný prach nebo kovový prášek, žádný radioaktivní materiál, žádný elektromagnetický šum, nehořlavé předměty.
- Zabraňte tomu, aby zařízení M om pracovalo v nestabilních podmínkách proudu.
- Uživatelé musí zvýšit vzduchový spínač se schopností ne méně než 10A/250V v M ont výkonu robota.

Poznámka: Pokud chcete kolaborativního robota zavěsit nebo nainstalovat, kontaktujte nás.

# 1.3.3.4.4 Kapacita podlahového nosiče

Instalace robota na pevný povrch, povrch by měl být dostatečný, aby vydržel hmotnost robotického ramene alespoň 5krát, a povrch nesmí být vibrovaný.

#### 1.3.3.4.5 Maximální platné zatížení

Maximální přípustné platné zatížení robotického ramene závisí na těžišti. Když se těžiště břemene vzdálí, bude zatížení, které robot ponese, menší.

Obrázek 2.4-15 Křivka zatížení modelu M 3 spolupracujícího robota Obrázek 2.4-16 Křivka zatížení modelu M 5 spolupracujícího robota Obrázek 2.4-17 Křivka zatížení modelu M 10 spolupracujícího robota Obrázek 2.4-18 Křivka zatížení modelu M 16 spolupracujícího robota Obrázek 2.4-19 Křivka zatížení modelu M 20 spolupracujícího robota

# 1.3.3.5 Připojení ovládání

# 1.3.3.5.1 Rozhraní ovladače

Tato řada oMobotů využívá jednofázový zdroj střídavého proudu TN-S 220 V a zařízení je dodáváno s 5metrovým napájecím kabelem. Tříkolíkový konec zástrčky se zasouvá do zásuvky střídavého napětí 220 V na místě a robot je elektricky uzemněn.

- Jmenovité vstupní napětí: 6A/220VAC
- Jmenovité výstupní napětí: 48V/21A
- Počet fází: jednofázové

# <u>M, verze 1.0.0</u>

- Frekvence: 50 Hz
- Výstupní zkratová odolnost: 48V/22A











Varování: Před zapojením se ujistěte, že je zdroj napájení vypnutý, a zavěste vedle něj bezpečnostní výstražnou ceduli.

Vnější kabeláž této řady řídicích systémů oMobotic arm je připojena pomocí zásuvných a rychle instalovatelných konektorů. Panel zapojení kolaborativního robota je znázorněn na obrázku 2.5-1.

- Ujistěte se, že je napájecí kabel 220 V připojen do zásuvky s vypnutým tlačítkem napájení ovládacího panelu (tlačítko je otočeno na 0) (vstupní napětí při plném zatížení je 6A/220VAC~7A/210VAC).
- Připojte kabel pro přetížení těla robota k rozhraní pro přetížení ovládací skříňky.
- Vložte leteckou zástrčku tlačítkového boxu do rozhraní ovládacího boxu výukového zařízení.
- Otvory pro odvod tepla na obou stranách ovládací skříňky by měly být od sebe vzdáleny nejméně 15 cm.
- Na M ont ovládací skříňky (uživatelský kovový stůl, tlačítko napájení, těžké zatížení a výukový závěsný kabel), lázně
- Ovládací skříňka je ve výšce 0,6-1,5 m nad zemí.
- Nedovolte uživatelům, aby si sami vyměňovali napájecí kabely.

Obrázek 2.5-1 Schéma zapojení robota

# 1.3.3.5.2 Panel I/O řídicí jednotky

Pomocí I/O v ovládací skříňce můžete ovládat různá zařízení, včetně tlačítka stop pneumatického relé, PLC a těsného koncového zařízení. Na obrázku 2.5-2 je zobrazena skupina elektrických rozhraní ovládací skříňky. Obrázek 2.5-3 ukazuje skupinu elektrického rozhraní, kterou lze snadno vyrobit z ovládací skříňky.

Obrázek 2.5-2 Schéma elektrického rozhraní ovládací skříně

Obrázek 2.5-3 Schéma elektrického rozhraní snadno vyrobitelné ovládací skříňky

.



Power communication	Universal d	ligital input	Configurable	digital inputs	Security	Universal d	ligital output	Configurable	digital output	Encoder	Analog
ex24V	GND	GND	GND	GND	EIO+	24V	24V	24V	24V	A1-	GND
exGND	DIO	DI4	CIO	CI4	E10-	DO0	DO4	COO	CO4	A1+	AIO
24V	GND	GND	GND	GND	EI1+	24V	24V	24V	24V	B1-	GND
GND	DI1	DI5	CI1	CI5	E11-	DO1	DO5	CO1	CO5	B1+	Al1
5V	GND	GND	GND	GND	SI0+	24V	24V	24V	24V	A2-	GND
GND	DI2	DI6	CI2	CI6	SI0-	DO2	DO6	CO2	CO6	A2+	AO0
485-B	GND	GND	GND	GND	SI1+	24V	24V	24V	24V	B2-	GND
485-A	DI3	DI7	CI3	C17	511-	DO3	DO7	CO3	C07	B2+	AO1
User periphera M8 Waterpr	al interface	ř.						etwork	LAN netwo	ork (spare)	USB-2.0 interface
	then the electrical e powered off. tuations that may erious damage to e circumstances.	l interface of the o y cause danger ar equipment; Mattu sometimes have	control cabinet is nd, if not avoided ers marked with the possibility of	wired, the contro I, can lead to per this symbol may, significant conse	l cabinet must sonal injury or depending on quences.		2 J14	J8: J12: J14:	I: Analog cur V: Analog voi I: Analog cur V: Analog voi	rent input ( tage input ( rent output ( tage output (	0-20mA) 0-10V) (0-20mA) (0-10V)

Name.	Contro	ller				Mar	nufacture	er: FAIF	R INNO	<b>ATION</b>	(Suzho	u) Robot	System	n Co.,LTI	)	
Model	FRC10	D-AC					Addres	s: 209	Zhuyua	in oad,H	ligh-teo	h Zone,	Suzou (	City, Jiang	jsu Prov	vince
nperature	0-45°C							588	8 Zunxi	an Road	,High-t	ech Zon	e,Zibo (	City,Shar	ndong F	rovine
Class	IP54					81	lelephor	ie: 051	2-68562	2005				_		
	220VA	C/10A/S	Single P	hase/50	HZ	Seria	I Numb	er: 「						<b>−</b> ?		
13	6000W	/48VDC	/41A		Da	te Of P	roductio	n:						Ê		
								-						_		
2 3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
DV EIO-1	EI1-1	SI0-1	511-1	EDM-	EST0-1	EST1-1	485-B0	485-B1	5 V	B1+	B1-	B2+	B2-	GNDA	A00	A01
V EIO-2	EI1-2	510-2	511-2	EDM+	ESTO-2	EST1-2	485-A0	485-A1	GND	A1+	A1-	A2+	A2-	GNDA	AI 0	AI 1
2 3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
01 D02	D03	E-24V	D04	D05	D06	D07	E-24V	C00	C01	C02	C03	E-24V	C04	C05	C06	C07
1 DI 2	DI 3	E- OV	DI 4	DI 5	DI 6	DI 7	E- 0V	CI 0	CI 1	CI 2	CI 3	E- 0V	CI 4	CI 5	CI 6	CI 7
																P
	Model nperature Class 2 3 2 8 2 8 2 8 2 8 2 8 2 8 2 8 2 8 2 8 2 8	Model: FRC10 nperature: 0-45°C Class: IP54 : 220VA : 6000W 2.34 W EI0-1 EI1-1 V EI0-2 EI1-2 2.34 11 D02 D03 1 D12 D13	Model: FRC100-AC nperature: 0-45°C Class: IP54 : 220VAC/10A/S : 6000W/48VDC 2.345 VED-1 EI1-1 SI0-1 VED-2 EI1-2 SI0-2 2.345 1.002 D03 E-24V 1.012 D13 E- 0V	Model: FRC100-AC nperature: 0-45°C Class: IP54 : 220VAC/10A/Single P : 6000W/48VDC/41A 2 3 4 5 6 10 EI0-1 EI1-1 SI0-1 SI1-1 V EI0-2 EI1-2 SI0-2 SI1-2 2 3 4 5 6 11 D02 D03 E-24V D04 1 D12 D13 E- 0V D14	Model: FRC100-AC nperature: 0-45°C Class: IP54 : 220VAC/10A/Single Phase/50 : 6000W/48VDC/41A 2.34567 V EI0-1 EI1-1 SI0-1 SI1-1 EDM- V EI0-2 EI1-2 SI0-2 SI1-2 EDM+ 2.34567 10 D02 D03 E-24V D04 D05 1 D12 D13 E- 0V D14 D15	Model: FRC100-AC nperature: 0-45°C Class: IP54 : 220VAC/10A/Single Phase/50HZ : 6000W/48VDC/41A Da 2.345678 VEI0-1 EI1-1 SI0-1 SI1-1 EDM- EST0-1 VEI0-2 EI1-2 SI0-2 SI1-2 EDM+ EST0-2 2.3456788 10 D02 D03 E-24V D04 D05 D06 1 D12 D13 E- 0V D14 D15 D16	Model: FRC100-AC nperature: 0-45°C Class: IP54 : 220VAC/10A/Single Phase/50HZ Serie : 6000W/48VDC/41A Date Of P 2 3 4 5 6 7 8 9 10 EI0-1 EI1-1 SI0-1 SI1-1 EDM- EST0-1 EST1-1 V EI0-2 EI1-2 SI0-2 SI1-2 EDM+ EST0-2 EST1-2 2 3 4 5 6 7 8 9 11 D02 D03 E-24V D04 D05 D06 D07 1 D12 D13 E- 0V D14 D15 D16 D17	Model: FRC100-AC Addres nperature: 0-45°C Telephon : 220VAC/10A/Single Phase/50HZ Serial Numbe : 6000W/48VDC/41A Date Of Production 2 3 4 5 6 7 8 9 10 W EI0-1 EI1-1 SI0-1 SI1-1 EDM- EST0-1 EST1-1 485-80 V EI0-2 EI1-2 SI0-2 SI1-2 EDM+ EST0-2 EST1-2 485-A0 2 3 4 5 6 7 8 9 10 1 D02 D03 E-24V D04 D05 D06 D07 E-24V 1 D12 D13 E- 0V D14 D15 D16 D17 E- 0V	Model:       FRC100-AC       Address:       209         nperature:       0-45°C       588         Class:       IP54       Telephone:       051         :       220VAC/10A/Single Phase/50HZ       Serial Number:       Date Of Production:	Model:       FRC100-AC       Address:       209 Zhuyua         apperature:       0-45°C       5888 Zunxia         Class:       IP54       Telephone:       0512-68562         :       220VAC/10A/Single Phase/50HZ       Serial Number:	Model:       FRC100-AC       Address:       209 Zhuyuan oad, F         nperature:       0-45°C       5888 Zunxian Road         Class:       IP54       Telephone:       0512-68562005         :       220VAC/10A/Single Phase/50HZ       Serial Number:       [         :       6000W/48VDC/41A       Date Of Production:       [         2       3       4       5       6       7       8       9       10       11       12       13         IV       EI0-1       EI1-1       SI0-1       SI1-1       EDM-       EST0-1       EST1-1       485-80       485-A1       GND       A1+         V       EI0-2       EI1-2       SI0-2       SI1-2       EDM+       EST0-2       EST1-2       485-A0       485-A1       GND       A1+         2       3       4       5       6       7       8       9       10       11       12       13         10       D02       D03       E-24V       D04       D05       D06       D07       E-24V       C00       C01       C02         11       D12       D13       E-0V       D14       D15       D16       D17       E-0V       C10       C	Model:       FRC100-AC       Address:       209 Zhuyuan oad,High-tec         nperature:       0-45°C       5888 Zunxian Road,High-tec         Class:       IP54       Telephone:       0512-68562005         :       220VAC/10A/Single Phase/50HZ       Serial Number:          :       6000W/48VDC/41A       Date Of Production:          :       :       6000W/48VDC/41A       Date Of Production:          :       :       :       :       :       :         :       :       :       :       :       :         :       :       :       :       :       :         :       :       :       :       :       :         :       :       :       :       :       :         :       :       :       :       :       :         :       :       :       :       :       :         :       :       :       :       :       :         :       :       :       :       :       :         :       :       :       :       :       :       :         :       :       :	Model:       FRC100-AC       Address:       209 Zhuyuan oad,High-tech Zone, 5888 Zunxian Road,High-tech Zone, 5888 Zun	Model:       FRC100-AC       Address:       209 Zhuyuan oad, High-tech Zone, Suzou G         nperature:       0-45°C       S888 Zunxian Road, High-tech Zone, Suzou G         Class:       IP54       Telephone:       0512-68562005         :       220VAC/10A/Single Phase/50HZ       Serial Number:	Model:       FRC100-AC       Address:       209 Zhuyuan oad,High-tech Zone,Suzou City,Jiang 5888 Zunxian Road,High-tech Zone,Zibo City,Shar Class:         Imperature:       0-45°C       Telephone:       0512-68562005         Class:       IP54       Telephone:       0512-68562005         :       200VAC/10A/Single Phase/50HZ       Serial Number:	Model:       FRC100-AC       Address:       209 Zhuyuan oad,High-tech Zone,Suzou City,Jiangsu Prov 5888 Zunxian Road,High-tech Zone,Zibo City,Shandong P         Class:       IP54       Telephone:       0512-68562005         :       200VAC/10A/Single Phase/50HZ       Serial Number:

# 1.3.3.5.3 Skupina síťových rozhraní RJ45

Adresa skupiny síťových rozhraní v řídicí skříňce je znázorněna na obrázku 2.5-3. Všimněte si, že graf odpovídá pořadí adres vnitřního síťového portu řídicí skříňky a výchozí port robota je zakázán M om insertion. Síťový port uživatele lze použít ke komunikaci s kamerou a dalšími zařízeními. IP adresa je 192.168.57.2. Rozhraní tlačítkové skříňky je ve výchozím nastavení nastaveno na fakultní řídicí port a adresa IP je 192.168.58.2. Použijte síťový kabel pro propojení rozhraní tlačítkové skříňky a počítače. IP adresa počítače je nastavena na 192.168.58.10 nebo na stejný síťový segment jako on. Můžete přistupovat na stránku osciloskopu. Snadno vyrobitelné ovládací boxy pro přístup ke stránkám osciloskopu přes síťový port připojovacího tlačítkového boxu.

EtherCAT:robot	default port	
Teaching:	192. 168. 58. 2	$\left( \begin{array}{c} \end{array} \right)$
User Port:	192. 168. 57. 2	$\left( \begin{array}{c} \end{array} \right)$

Obrázek 2.5-4 Významové schéma skupiny síťových rozhraní

#### 1.3.3.5.4 Koncová deska

Pomocí I/O a komunikačního rozhraní 485 koncového panelu můžete ovládat různá zařízení, včetně pneumatických relé, PLC a tlačítek nouzového zastavení. Rozdělení patic PIN a jejich vysvětlení je uvedeno na obrázku 2.5-4. Model konektoru I/O je osmijádrový konektor M12.



Obrázek 2.5-5 Schéma koncové verze elektrického rozhraní

# 1.3.3.5.5 Pozemek

1. Ovládací skříňka se nachází na kombinovaném šroubu M4 v levém horním rohu vypínače, jak je znázorněno na obrázku 2.5-5.

Obrázek 2.5-6 Demonstrační schéma ovládací skříňky

2. Těleso je umístěno na pravé straně základny základny, jak je znázorněno na obrázku 2.5-6.

Obrázek 2.5-7 Dračí schéma tělesa Samotný ochranný

vodič by neměl mít menší průřez než:

- 2,5 mm<sup>2</sup> mědi nebo 16 mm<sup>2</sup> hliníku, pokud je zajištěna ochrana proti mechanickému poranění (drátěné potrubí, potrubí atd.).
- 4 mm<sup>2</sup> měď nebo 16 mm<sup>2</sup> hliník, pokud není zajištěna ochrana proti mechanickému poškození.



outlet of the base Grounding position

# 1.3.3.5.6 Společné specifikace všech digitálních I/O

V této části jsou uvedeny elektrické specifikace následujícího digitálního vstupu/výstupu 24 V ovládací skříňky:

- Bezpečnostní vstupy/výstupy
- Univerzální digitální množství I/O

Roboty musí být instalovány v souladu s elektrickými specifikacemi.

Konfigurací rozhraní "Power Communication" můžete k napájení digitálních I/O použít interní nebo externí 24V napájení. Výše uvedené dvě svorky (EX24V a EXON) v rozhraní jsou 24V a zem externího napájení a níže uvedené dvě svorky (24V a GND) jsou 24V a zem interního napájení. Ve výchozí konfiguraci se používá interní napájení, jak je znázorněno na obrázku 2.5-7.



Obrázek 2.5-8 Schéma komunikace napájení 01

Pokud je výkon zátěže velký, můžete připojit externí zdroj napájení podle obrázku 2.5-8.



Obrázek 2.5-9 Schéma komunikace napájení 02

Elektrické specifikace interního a externího napájení jsou uvedeny v tabulce 2.5-1. Interní a externí elektrické specifikace:

Tabulka 2.5-1 Elektrické specifikace interního a externího zdroje napájení

Terminál	Parametr	Min- inum	Ty i- cal	Max- i- mum	Jed notl a
Interní napájení 24 V [ex24V -exGND] [ex24V -exGND]	Napětí Aktuální	23 0	24 -	25 2	V A
Interní napájení 24 V [24V- GND] [24V- GND]	Napětí Aktuální	23 0	24 -	25 1.5	V A

Elektrické specifikace digitálních I/O jsou uvedeny v tabulce 2.5-2 Elektrické specifikace digitálních I/O:

Tabulka 2.5-2 Elektrická specifikace digitálních I/O

Terminál	Parametr	Min- inum	Ty i- cal	Max- i- mum	Jed notl a
Digitální výstup [COx/BOx] [COx/DOx]	Aktuální Pokles tlaku Svodový proud	0 0 0	- -	1 0.5 0.1	A V mA
[COx/DOx]	funkce	-	NP	-	Тур
Digitální výstup [EIx/SIx/CIx/DIx] [EIx/SIx/CIx/DIx] [EIx/SIx/CIx/DIx]	OFF NA Proud (11~30)	-3 11 2	-	5 30 15	V V mA
[EIx/SIx/CIx/DIx]	funkce	-	NP	-	Тур

# 1.3.3.5.7 Bezpečnostní vstupy/výstupy

Tato část popisuje elektrické specifikace bezpečnostních I/O a musí být v souladu s obecnými elektrickými specifikacemi v části 1.6.6.

Bezpečnostní zařízení a vybavení musí být instalováno v souladu s bezpečnostním popisem a hodnocením rizik, viz kapitola 1.1. Všechny bezpečnostní I/O jsou párové (redundantní) a musí být uloženy jako dvě nezávislé větve. Jednotlivá selhání by neměla způsobit ztrátu bezpečnostní funkce.

Bezpečnostní vstupy/výstupy zahrnují nouzové zastavení a bezpečnostní zastavení. Vstup urgentního zastavení se používá pouze pro zařízení nouzového zastavení a vstup bezpečného zastavení pro různá bezpečnostní ochranná zařízení. Funkční rozdíly jsou uvedeny v tabulce 2.5-3:

#### Tabulka 2.5-3 Funkční rozdíl

	Nouzové zastavení	Bezpečná zastávka
Robot se přestane pohybovat	Ano	Ano
Kategorie zastávek	Kategorie 0	Kategorie 1
Provádění programu	Zastavte	Pauza
Napájení robota	Zavřít	Otevřít
Restartování	Manuální	Automatické nebo manuální
Četnost používání	InMequent	Často
Nutná reinicializace	Potřebujete	Zbytečné

#### Varování:

- Nepřipojujte bezpečnostní signál k PLC, které nemá správnou a bezpečnou úroveň. Pokud toto upozornění nedodržíte, může dojít k vážnému poškození nebo smrti, protože může být zakryta jedna z funkcí bezpečnostního zastavení. Signály bezpečnostního rozhraní musí být odděleny M om normální signály rozhraní I/O.
- Všechny I/O jsou redundantní -příbuzné I/O (dva nezávislé kanály). Dva kanály musí být vedeny odděleně, aby jediná porucha nezpůsobila bezpečnostní funkci.
- Před uvedením robota do provozu je nutné ověřit bezpečnostní funkci nouzového zastavení (robot je zapnutý, stiskne se tlačítko nouzového zastavení, robot se odpojí, vypne se napájení, otočí se tlačítko nouzového zastavení, zapne se napájení a robot se znovu zapne). Bezpečnostní funkce je třeba pravidelně testovat.
- Instalace robota by měla být v souladu s těmito specifikacemi. V opačném případě může dojít k vážnému poškození nebo smrti, protože může dojít k ukončení funkce bezpečnostního zastavení.

V následujících kapitolách jsou uvedeny příklady použití bezpečnostních vstupů a výstupů.

Výchozí bezpečnostní konfigurace Když robot opustí továrnu, má výchozí konfiguraci. Může být provozován bez dalších bezpečnostních zařízení. Viz obrázek 2.5-9.

Obrázek 2.5-10 Schéma bezpečnostní ochrany 01

**Připojení tlačítka nouzového zastavení** Ve většině aplikací je třeba použít jedno nebo více dalších tlačítek nouzového zastavení. Viz obrázek 2.5-10.

Obrázek 2.5-11 Schéma bezpečnostní ochrany 02

**Připojte bezpečnostní stop tlačítko** Příkladem bezpečného stop zařízení je dveřní spínač, který robota zastaví při zapnutí dveří. Viz obrázek 2.5-11.

Obrázek 2.5-12 Schéma bezpečnostní ochrany 03







## 1.3.3.5.8 Univerzální digitální množství I/O

Tato část popisuje elektrické specifikace obecných digitálních I/O a musí být v souladu s obecnými elektrickými specifikacemi v části 1.6.6.

Obecné digitální množství I/O lze použít k řízení relé, elektromagnetických ventilů a dalších zařízení nebo k interakci s jinými PLC.

# Řídicí zátěž digitálního výstupního množství

Tento příklad ukazuje, jak připojit výstup digitální veličiny pro řízení zátěže, viz obrázek 2.5-12.



Obrázek 2.5-13 Schéma velkého digitálního výstupu veličiny 01

#### 1.3.3.5.9 Digitální vstup M om tlačítko

Následující příklad ukazuje, jak připojit jednoduché tlačítko ke vstupu digitální veličiny.



Obrázek 2.5-14 Schéma velkého digitálního výstupu veličiny 02

# 1.3.3.5.10 Interakce s jinými zařízeními nebo PLC



Následující příklad ukazuje, jak komunikovat s jinými zařízeními nebo digitálním vstupem a výstupem PLC.

Obrázek 2.5-15 Interaktivní diagram s jinými zařízeními nebo PLC

# 1.3.3.5.11 Simulace I/O

Tabulka 2.5-4 Simulační proudové napětí

Terminál	Parametr	Min- inum	Ty i- cal	Max- i- mum	Jed notl a
Analogový proudový vstup [AIx-END] [AIx-END] [AIx-END]	Rozlišení proudové impedance	0 - -	- 500 12	20 - -	mA ohm bit
Analogový napěťový vstup [AIx-END] [AIx-END] [AIx-END]	Rozlišení napěťové impedance	0 - -	- 500 12	10 - -	V Kohrr bit
Analogový proudový vstup [AOx-END] [AOx-END] [AOx-END]	Rozlišení aktuálního napětí	0 0 -	- - 12	20 10 -	mA V bit
Analogový napěťový vstup [AOx-END] [AOx-END] [AOx-END] [AOx-END]	Napětí Proud Rozlišení impedance	0 0 -	- - 100 12	10 20 -	V mA ohm bit

Simulační I/O slouží k nastavení nebo měření napětí (0-10 V) nebo proudu (0-20 mA) jiných zařízení.

Pro dosažení vysoké přesnosti se doporučují následující metody.

- Zařízení a ovládací skříňka používají stejnou zem (GND).
- Používejte stínicí kabely nebo kroucená vedení.

Následující příklad ukazuje, jak používat analogové I/O.

# Použití analogového výstupu

Následující příklad demonstruje použití analogového výstupu pro řízení dopravníkového pásu.

Obrázek 2.5-16 Schéma výstupu simulace



#### Použití analogového vstupu

Následující příklad slouží k demonstraci simulačního vstupního připojení simulačního senzoru.





#### 1.3.3.6 Demonstrace a ukončení LED

Robotický osciloskop může k přístupu a ovládání robota používat počítač nebo tablet. Způsob připojení lze vysvětlit v části 1.6.3. Kromě toho mohou uživatelé používat také náš ost at ní M -HMI.

# 1.3.3.6.1 Úvod do pole s tlačítkem

První verze tlačítkového pole

Obrázek 2.6-1 První verze tlačítkového pole

Spínač nouzového zastavení: Po stisknutí spínače nouzového zastavení přejde robot do stavu nouzového zastavení.

Typ-c:Připojte port webového osciloskopu.

Tlačítko 1: Krátce stiskněte přepínač automatického/ručního režimu, dlouze stiskněte a vstupte do režimu přetahování nebo jej ukončete.

Tlačítko 2:Krátkým stisknutím záznamu zobrazíte výukový bod, dlouhým stisknutím vstoupíte do stavu bez demonstrátoru.



# **Tlačítko 3:** Spuštění/zastavení spuštěného programu. Druhá verze tlačítkového pole



Obrázek 2.6-2 Druhá verze pole tlačítek

Spínač nouzového zastavení: Po stisknutí spínače nouzového zastavení přejde robot do stavu nouzového zastavení.

Spuštění/zastavení: Spuštění/zastavení

spuštěného programu. Ethernet:

Připojení k webovému osciloskopu.

Vypnout: Není povoleno.

Bod záznamu: Zaznamenejte bod výuky.

Režim výuky: Vstupte/vyjděte ze stavu výukového přívěsku.

Pracovní režim: Přepínač automatického/manuálního režimu.

Režim přetahování: Vstupte do režimu přetahování nebo jej ukončete.

#### 1.3.3.6.2 Úvod k přívěsku M-HMI Teach



Obrázek 2.6-3 Výukový přívěsek M -HMI M ont Obrázek 2.6-4 Zadní strana výukového přívěsku M -HMI

Displej: Dotykové ovládání a rozhraní displeje výukového přívěsku.

Klávesa Start: Spuštění programu.

Klávesa Stop: Zastaví aktuálně spuštěný

program. Tlačítko F4:Vstup na klávesnici pro

vyvolání kláves. Tlačítko Kloub: Kloubní

uzel robota.

Tři -bit enable: Roboti povolující manuální režim

**Spínač nouzového zastavení:** Po stisknutí spínače nouzového zastavení přejde robot do stavu nouzového zastavení.

Tlačítko režimu: Otáčením tlačítka přepnete na automatický režim.

\_



 $\Gamma^{\ast}$ 

#### 1.3.3.6.3 Konec definice LED

Funkce	Barva LED
Když není navázána komunikace	Střídavě "Vypnuto", "Červená", "Zelená" a "Modrá".
Automatický režim	Modrá dlouhá světlá
Manuální režim	Zelená dlouhá světlá
Režim přetahování	Bílá azurová dlouhá světlá
Bod záznamu tlačítkového pole (pouze při použití tlačítkového pole)	Fialová dvakrát blikne
Spustit běh (pouze při použití tlačítkového pole)	Azurově modrá dvakrát zabliká
Zadejte stav neshodného tlačítkového pole (pouze v případě, že se jedná o us- v poli tlačítek)	Modrá dvakrát zabliká
Zastavení provozu (pouze při použití tlačítkového pole)	Červená dvakrát blikne
Hlášení chyb (pouze při použití tlačítkového pole)	Červená dlouhá světlá
Kalibrace nuly dokončena	Bílá azurová třikrát blikne
Povolit	Žlutá dvakrát blikne

Tabulka 2.6-1 Tabulka definic koncových LED diod

# 1.3.4 Rychlé spuštění

#### 1.3.4.1 Instalace ramene robota a ovládacího panelu

Nainstalujte rameno robota a ovládací skříňku podle bodů 2.5 a 2.6 v části Instalace hardwaru v kapitole 2.

- Vyjměte robotické rameno a pomocí 4 šroubů M8 s pevností nejméně 8,8 úrovně namontujte robotické rameno. Rameno robota nainstalujte na pevný a nárazuvzdorný povrch. Pokud je upevněno pomocí hliníkové desky, tloušťka hliníkové desky není menší než 16 mm. Pokud je připevněna železná deska, tloušťka železné desky není menší než 8 mm;
- Umístěte ovládací skříňku na nožičky;
- Připojte ontologii robotického ramene k rozhraní pro těžké zatížení řídicí jednotky;
- Zapojte leteckou zástrčku tlačítkové skříňky do příkladného rozhraní ovládací skříňky. Pokud jste vybaveni verzí teentru s dotykovou obrazovkou, musíte také použít datový kabel obou konců jako rozhraní typu C, vložte typ tlačítkové krabičky a instruktor, resp. Rozhraní -c;
- Ujistěte se, že je tlačítko napájení ovládací skříňky zavřené (tlačítko je nastaveno na 0), abyste mohli připojit napájecí kabel 220 V do zásuvky;
- Zapojte zástrčku ovládací skříňky napájení.

Varování: Pokud je robot umístěn na silném povrchu, může dojít k jeho pádu a poškození.

# 1.3.4.2 Demonstrace spuštění řídicího robota

Řídicí skříňka je připojena k rameni robota, výukové skříňce a k fyzickému elektrickému vstupnímu/výstupnímu terminálu jakéhokoli periferního zařízení. Chcete-li napájet rameno robota, musíte otevřít ovládací skříňku.

- Stisknutím tlačítka napájení ovládací skříňky otevřete ovládací skříňku;
- Po spuštění robota je v tuto chvíli robot v manuálním režimu a není aktivován. Pokud je třeba robota ovládat v manuálním režimu, je třeba stisknutím tří bitů na osciloskopu vypnout (pustit) OFF (stisknout) jako spínač jako spínač Drag nebo ovládat pohyb robota, když jste na svém stavu.
- Pokud nepotřebujete robota ovládat v manuálním režimu, můžete pomocí klíčového spínače přepnout pracovní režim robota na tlačítko pro přepínání: automatický, manuální, vlastní;
- Při přepínání robota do ručního stavu zkontrolujte, zda uvnitř a vně bezpečnostního prostoru nedochází k abnormálním odchylkám, a pečlivě ovládejte provoz stroje;
- Při přepnutí robota do automatického stavu zkontrolujte bezpečnostní opatření, vraťte se do normálního stavu a
  pečlivě provozujte stroj;
- Pokud osciloskop nelze normálně otevřít, zkontrolujte, zda je připojení zařízení v pořádku.

#### 1.3.4.3 Ovládání pohybu robota tlačítkovou krabičkou

Viz 2.7.3 Instalace hardwaru v kapitole 2. Koncová LED definice konce pro ovládání robota

#### 1.3.4.3.1 Není spárováno s výukovým přívěskem

2<sup>n</sup>

• Krok 1Zapněte vypínač ovládací skříňky robota, spusťte robota, před spuštěním robota počkejte, až se koncová LED dioda dlouho zobrazí zeleně, jak je znázorněno na obrázku 3.3-1.



Obrázek 3.3-1 Zelené schéma koncové LED diody

• Krok2Dlouhým stisknutím tlačítka "button 2" přejděte do režimu bez výukového přívěsku a koncová LED dioda třikrát modře zabliká, jak je znázorněno na obrázku 3.3-2.

r



Obrázek 3.3-2 schéma modré koncové LED diody

- Krok3Dlouze stiskněte tlačítko "tlačítko 1" Přepněte robota do režimu tažení. V této chvíli svítí koncová LED bíle a modře, jak je znázorněno na obrázku 3.3-3. Do libovolné polohy mobilního robota dlouhým stisknutím tlačítka "button 1" ukončete režim přetahování, krátkým stisknutím tlačítkového pole "button 2" zaznamenáte bod P1, koncová LED třikrát fialově zabliká, jak je znázorněno na obrázku 3.3-4.
- Krok4Mobilní robot, krátce stiskněte tlačítko box "tlačítko 2" záznam P2 bod, konec LED fialová bliká třikrát, jak je znázorněno na obrázku 3.3-4.



• Krok5Dlouhým stisknutím tlačítka "tlačítko 1" ukončete režim přetahování. V tuto chvíli se jedná o ruční režim. Svítí

koncová LED dioda

r



zelená, jak je znázorněno na obrázku 3.3-5. Krátkým stisknutím "Key 1" přepnete robota do automatického režimu. V tomto okamžiku svítí koncová LED modře, jak je znázorněno na obrázku 3.3-6.

• Krok6Krátké stisknutí tlačítka "button 3" spustí program, koncová LED dvakrát modře zabliká, jak je znázorněno na obrázku 3.3-6.



Obrázek 3.3-5 schéma koncové zelené LED diody

Obrázek 3.3-6 schéma koncové modré LED diody

• Krok7Krátkým stisknutím tlačítka v poli "button 3" zastavte spuštění programu a červená LED dioda na konci třikrát zabliká, jak je znázorněno na obrázku 3.3-7.

Obrázek 3.3-7 schéma červené koncové LED diody



i.

i"

1



#### 1.3.4.3.2 Shoda osciloskopie

- Krok1Před spuštěním robota počkejte, až přestane blikat zelená koncová LED dioda.
- Krok2Otevřete osciloskop a vstupte do rozhraní pro úpravu programu.
- Krok3Výběr prázdné šablony pro vytvoření nového souboru programu.
- Krok4Krátké stisknutí tlačítka boxu Tlačítko 1 Přepněte robota do manuálního režimu. V tomto okamžiku svítí koncová LED zeleně.
- Krok5Dlouhým stisknutím tlačítka boxu 1 přepněte robota do režimu tažení. V této době je koncová LED bílá a zelená, mobilní robot do libovolné polohy, krátkým stisknutím tlačítka box tlačítka 2 zaznamená P1, koncová LED třikrát fialově zabliká, ručně přidejte "PTP: P1 "Pokyny do souboru programu.

🗅 test.lua	
1 ↔ PTP:P1,100	

Obrázek 3.3-8 zaznamenejte a přidejte některé P1

• Krok6Mobilní robot, krátké stisknutí tlačítka boxu tlačítka 2 zaznamená bod P2, koncová LED dioda třikrát fialově zabliká a ručně přidá do programu instrukci "PTP: P2".

Obrázek 3.3-9 záznam a přidání některých P2

- Krok7Uložení obsahu souboru programu.
- Krok8Dlouhým stisknutím tlačítka boxu 1 ukončete režim přetahování. V tuto chvíli se jedná o ruční režim. Kontrolka ukončení svítí zeleně. Krátkým stisknutím tlačítka box button 1 přepněte robota do automatického režimu. V tomto okamžiku je koncová LED dioda modrá.
- Krok9Krátké stisknutí tlačítka box tlačítko 3 Spustit program, konec LED modře blikne dvakrát.

🗅 test.lua

1 ↔ PTP:P1,100

2 ↔ PTP:P2,100

# 1.3.4.4 Dempermaster řídí pohyb robota

Klikněte na tlačítko "Promotional Simulation" v nabídce první úrovně na levé straně osciloskopu a klikněte na podnabídku "Promotional Simulation".

-menu "Demonstrace programu" pro vstup do rozhraní demonstrace programu. Rozhraní realizuje především zápis a úpravu výukového programu robota.

Po kliknutí na tlačítko s ikonou "Nový" uživatel pojmenuje soubor a vybere šablonu jako obsah nového souboru. Klepnutím na nově vytvořený vytvoří úspěšný a otevře programový soubor.

a ≡	(		Stopped tooi1 wobj1 exa	ixis1 100 🖢		8
🕸 Initialize <	📹 🖻 🕁 🛨	,ᄜᄜᆝᆃᇾᅊᇳᅀᅙ│	Cperation& Status 360	* Free Mounting Fb	ked Mounting	
Teaching	6 N	[] left_pattern_b lua	☑	Robot Pose		(.its.)
Program Teaching	PTP LIN	1→ total_start_time = GetSy		11.8	.p.6 .r	Joints
Graphical Program	ARC Circle	2⊷ layer_index = GetSysVa	━ −०− 🕀 📖	J4 : 0	J5:0 J6	:0
Manage Teaching	8 8	3⇔ box_index = 0	━−━ 🛟 📖	<b>X</b> 10	Y:0 Z	TCP
-A⊶ Status <	Spiral N-Spiral	4→ SetSysVarValue(s_var_5	Prefix	RX: 0	RY:0 RJ	Z: 0
🔠 Auxiliary <	Spline N-Spline	5⊷left B: box id = 1	Point name Add	Toci		FT
🗙 Settings	D A 1/0	$6 \rightarrow PTP(palletizing_TESTgri$	Sensor	Fx : 0 Tx : 0	Fy:0 Fz Ty:0 Tz	::0 ::0
		7↔ SetAuxDO(4,1,0,0)	Point name Add	Act_State:		
	TPD ToolList	8⊷ box_index = GetSysVar	Base coord. OFF	Num : 0		Line-Num
		9→ SetSysVarValue(s_var_5	Tool coord. OFF Workpiece ON			CtriBox
	Mode Var	10→ WaitMs(1000)	Ext. axis c. ON		001 0002 005 0005	0003
	Vy ↓ While If_Else	11↔ Lin(palletizing_TESTgri	Trajectory ON Import tool	d 800 80	001 CO2 005 CO6	0 CO3 0 CO7
	1 (4)				מות ווו	Ona

Obrázek 3.4-1 Ukázka schématu běhu výukového programu

Varování: Vaše hlava a trup se nesmí nacházet v dosahu (pracovní oblasti), kam robot dosáhne. Nevkládejte prsty do místa, které může robot uchopit.

#### Důležité:

- Nedovolte, aby se robot pohyboval směrem k vám nebo jiným objektům, protože by došlo k jeho poškození.
- Toto je pouze stručný návod, který vás naučí, jak snadno používat kolaborativní roboty M. Předpokladem této příručky je, že prostředí je bezpečné a neškodné a uživatelé jsou opatrní. Nezvyšujte rychlost ani zrychlení na výchozí hodnotu. Před vstupem robota do provozu se vždy provede posouzení rizik.

# 1.3.5 Výukový přívěsek software

#### 1.3.5.1 Základní informace

#### 1.3.5.1.1 Úvod

Software výukového závěsu je podpůrný software vyvinutý pro robota a běží na operačním systému výukového závěsu. Jeho hlavní funkce a technické vlastnosti jsou následující:

- Schopnost psát výukové programy pro roboty;
- Dokáže zobrazovat souřadnice polohy robota v reálném čase, simulovat fyzického robota ve třech rozměrech a řídit jeho pohyb;
- Může realizovat jednoosé inchingové a vazební operace každé osy robota;
- Možnost zobrazit stav řídicího IO;
- Uživatelé mohou měnit hesla, zobrazovat systémové informace atd.

#### 1.3.5.1.2 Spustit software

- 1. Zapněte ovládací skříňku;
- 2. Učitelský přívěšek otevře prohlížeč pro přístup k cílové webové stránce 192.168.58.2;
- 3. Zadejte uživatelské jméno a heslo a kliknutím na tlačítko Přihlásit se přihlaste do systému.

#### 1.3.5.1.3 Přihlášení uživatele a aktualizace oprávnění

#### Tabulka 4.1-1 Počáteční uživatel

Číslo zakázky	Počáteční uživatelské jméno	Heslo	Kód funkce
111	admin	123	1
222	MEenginer	222	2
333	PEenginer	333	3
444	programátor	444	4
555	operátor	555	5
666	monitor	666	6

Uživatelé (*pro správu uživatelů viz 4.10.2.1 Správa uživatelů*) jsou standardně rozděleni do šesti úrovní, administrátoři nemají žádná funkční omezení, operátoři a monitoři mohou používat malý počet funkcí, inženýři ME, inženýři PE&PQE a technici a vedoucí týmů mají některá funkční omezení, management Pro administrátory neexistují žádná funkční omezení. Konkrétní výchozí oprávnění ke kódům funkcí naleznete v části *4.10.2.2 Správa autorit*.

Přihlašovací rozhraní je znázorněno na obrázku 4.1-1 přihlašovací rozhraní.

Ā	
Star	(user name password
	Sign in
	Forgot password



Po úspěšném přihlášení systém načte model a další údaje a po načtení vstoupí na úvodní stránku.

# 1.3.5.2 Počáteční rozhraní systému

Po úspěšném přihlášení systém přejde do "úvodního rozhraní". Počáteční rozhraní ukazuje, že výukový přívěsek obsahuje především LOGO M INNOVATION a tlačítko pro návrat na počáteční stránku, lištu nabídek, tlačítko pro zvětšení lišty nabídek, oblast ovládání robota, oblast ovládání, stavovou oblast, oblast 3D simulace robota a oblast informací o poloze a IO, celkem osm oblastí. Jak je znázorněno na obrázku 4.2-1 schéma úvodního rozhraní systému.

a	≡	Stopped tool1 w	obj1 exaxis1	100	<b>W A</b>	8	8
Initialize	<	Operation&Status 360° Free Mounting Fixed	d Mounting				
🖻 Teaching	<	Joint Base Tool Wobj Move air.		Robot P	ose		<u>.</u>
<b>-∕∖-</b> Status	<	Eaxis IO TPD FT RCM					Joints
BB Auxiliary		Speed 100 %		J1 : 0 J4 : 0	J2 : 0 J5 : 0	JS : 0 JS : 0	
🗙 Settings		Acceleration 180 */s^2					TCP
				X :0	Y :0	Z : 0	101
		Single Multi		RX:0	RY:0	82:0	
				Tool			FT
				FX : 0	Fy : 0	F2 : 0	
				Act_State	1y : 0	12:0	
				-		L	ne-Num
				Num : 0			
						8	CtriBox
				0.000	0 D01	D02	D03
		Prefix		000	0001	002	CO3
		Point name Add		0.004	Onis	1000	TH3

Obrázek 4.2-1 Schéma počátečního rozhraní systému

# 1.3.5.2.1 Kontrolní oblast



```
efektNahrát a spustit výukový program
```



efektZastavení aktuálně spuštěného výukového programu



Poznámka: názevTlačítko Pauza / Pokračování efektPozastavení a obnovení aktuálního výukového programu

# 1.3.5.2.2 Stavový řádek

# Running

#### Poznámka:

stav namerobot

 $efekt Zastaveno-zastaven (B\check{e}h-Pauza-Pauza-PauzaP\check{r}et\acute{a}hnout-p\check{r}et\acute{a}hnout$ 



Poznámka:

nameČíslo souřadnicového systému nástroje

effectZobrazení čísla souřadného systému nástroje aktuální aplikace



Poznámka:

názevRychlost běhu v procentech

efektRychlost robota, když pracuje v aktuálním režimu.



Poznámka:

jménoRunn normálně

efektAktuální robot běží normálně



Poznámka:

nameError stav

effectV aktuální operaci robota došlo k chybě



#### Poznámka:

#### nameautomatický režim

Když je z a p n u t o globální nastavení rychlosti v manuálním a automatickém režimu a je zadána rychlost, globální rychlost se automaticky nastaví na zadanou rychlost.



#### Poznámka:

#### názevRežim výuky

efektRobot v režimu výuky, nastavte globální nastavení rychlosti manuálního režimu a automatického režimu a zadejte rychlost.



Poznámka:

#### nameDrag state

efektAktuální robot může přetáhnout



Poznámka: nameDrag state efektAktuální robot není tažný



nameStav připojení efektRobot připojen

Poznámka:



Poznámka: názevNení připojen stav effectRobot není připojen



Poznámka:

nameInformace o účtu

efektZobrazení uživatelského jména a oprávnění a odhlášení uživatele

# 1.3.5.2.3 Panel nabídek

Lišta nabídek je zobrazena v tabulce 4.2-1 Sloupce nabídek výukového přívěsku.

Tabulka 4.2-1 Sloupec nabídky učebního přívěsku

třída a	druhá úroveň		
Počáteční nastavení	Nastavení robota		
Výuková simulace	Konfigurace uživatelských periferií		
,	Program výuky		
	Grafické programování		
Informace o stavu	Řízení výuky		
	Systémový protokol		
Pomocná aplikace	Stavový dotaz		
1	Tělo robota		
	Databáze odborníků na svařování		
	Nastavení zabezpečení		
Nastavení systému	1		

#### 1.3.5.2.4 Provozní oblast

Nastavení IO naleznete v části *4.5.1 Nastavení I/O* v kapitole 4.5 Control Box I/O. Klouby, základny a další funkce naleznete v části *4.6 Obsluha robota*.

#### 1.3.5.3 3D simulace robota

#### 1.3.5.3.1 3D virtuální trajektorie a import modelu nástroje

Kreslení trajektoriePři spuštění výukového programu zapněte funkci kreslení trajektorie a 3D model robota bude popisovat trajektorii pohybu robota.

**Import modelu nástrojeKlikněte na** tlačítko "Importovat" a po importu modelu nástroje se model nástroje zobrazí na konci robota. V současné době jsou podporovány tyto formáty souborů modelu nástroje: STL a DAE.

━ ━ 🕂	
●	
● ●	
Prefix	
Point name	Add
Sensor •	]
Point name	Add
Base coord.	OFF
Tool coord.	OFF
Workpiece	ON
Ext. axis c	ON
Trajectory	ON
Import tool	Import

Obrázek 4.3-1 Kreslení virtuální trajektorie a import modelu nástroje
# 1.3.5.3.2 Zobrazení 3D vizualizace oMobotův souřadnicový systém

Ve virtuální 3D oblasti robota WebAPP vytvořte různé virtuální 3D souřadnicové systémy, jako příklad uveďme zobrazení základního souřadnicového systému, jak je znázorněno na obrázku níže. Mezi nimi je osa X červená, osa Y zelená a osa Z modrá.

**Základní souřadnicový systém V** základním souřadnicovém systému WebAPP je systémový robot ve výchozím nastavení zobrazen v trojrozměrné virtuální oblasti a pevná značka je ve spodním středu základny robota. Trojrozměrný virtuální souřadný systém základny lze zobrazit ručně.





Obrázek 4.3-2 Vypnutí a zapnutí zobrazení základního souřadného systému

**Souřadnicový systém** nástrojeZobrazení souřadnicového systému nástroje je ve výchozím nastavení povoleno a lze jej ručně vypnout. Po spuštění aplikace WebAPP a úspěšném přihlášení uživatele získejte název souřadnicového systému nástroje a odpovídající údaje parametrů aktuální aplikace a inicializujte aktuální souřadnicový systém nástroje.

Při použití jiných souřadnicových systémů nástroje během používání je třeba po úspěšném provedení příkazu pro použití souřadnicového systému nástroje nejprve vymazat stávající souřadnicový systém nástroje v 3D virtuální oblasti robota a poté přenést data parametrů nově použitého souřadnicového systému nástroje do 3D souřadnicového systému API pro generování souřadnicového systému nástroje a po dokončení generování se zobrazí v trojrozměrné virtuální oblasti robota.

a ≡	🥱   🕨 🔳 🕕 Stor	pped toolcoord1 wobj1 exaxis0 30 🖢 🛦 🕅 😰 😰
Initialize       Image: Status         Robot Settings       Image: Status         Peripheral Config       Image: Status         Image: Status       Image: Status         Image: Sta	of coord     Tool coordinate system settings       of coord     Current tool coordinate system       orkplete coordinate system     Coordinate system       of coordinate system     Colcoordinate system       of coordinate system     Colcoordinate system       of coordinate system     Colcoordinate system       of coordinate system     Coordinate system       of coordinate system     Coordinate system       of coordinate system     Stanse       oordinate System Settings     Coordinate System Settings       coordinate System Settings     Stanse       orting     Stanse       orting import     Stanse	Operation& Status 300° Free Mounting Flaed Mounting

Obrázek 4.3-3 Zobrazení souřadnicového systému nástroje

**Souřadnicový systém** obrobkuSouřadnicový systém obrobku je ve výchozím nastavení uzavřen a lze jej zobrazit ručně. Postup je konzistentní se souřadnicovým systémem nástroje.

a = (	🥱   🕨 🔳 🕕 sto	pped toolcoord1 wobj1 exaxis1 30 🖬 🛦 🕅 🏂 ⊗
<ul> <li>Initialize</li> <li>Initialize</li> <li>Robot Settings</li> <li>Peripheral Config</li> <li>Ext. tool coord</li> <li>Ext. tool coord</li> <li>Ext. tool coord</li> <li>Workpiece co.</li> <li>Teaching </li> <li>Ext. axis coord</li> <li>Status</li> <li>Status</li> <li>Soft limit</li> <li>End load</li> <li>Friction comp.</li> <li>Speed scaling</li> <li>IO fatering</li> <li>IO fatering</li> <li>Do config</li> <li>Config import.</li> </ul>	Workpiece coordinate system setting Current workpiece coordinate system Coordinate system name wobjcoord1 ~ X 300.000 Y 300.000 Z 300.000 RX 0.000 RY 0.000 R2 0.000 RX 0.000 RY 0.000 R2 0.000 Coordinate System Settings Modify Clear Apply	Operation & Status       350° Free Mounting       Fixed Mounting         J3       0       129 957       Robot Pose       127         J4       0       149 039       35       62.778       36       149 039         J5       0       0       95 01       62.778       36       149 039         J6       0       95 01       62.778       36       149 039       35       149 039         J6       0       0       95 01       95 01       149 039

Obrázek 4.3-4 Zobrazení souřadnicového systému obrobku

**Externí osový souřadnicový** systémVnější osový souřadnicový systém je ve výchozím nastavení vypnutý a lze jej ručně zapnout a zobrazit. Tento postup je konzistentní se souřadnicovým systémem nástroje.

	Q World enared	Federal and Andre Consellingto Federation Postform	
🖗 Initialize 🗸 🗸	• Trund coord.	Extended Axis Coordinate System Settings	Operation&Status   360° Free Mounting   Fixed Mounting
Robot Settings	Tool coord.	Current extended axis coordinate system	Robot Pose
Nobol Seconda	🔀 Ext. tool coord	Coordinate system name exaxis1 v	
Peripheral Config	Ø Workplece co	X [-541.834] Y [-102.999] Z [35.938	J4 😄 O 🛟 -149 038
Teaching 🔇	T Ext axis coord	RX -168.850 RY 24.005 RZ -59.162	J5 😄 -0 🛟 .42 778
⊨ Status 🔾	A Collision level	Ext. axis 1	J5 😑
🖁 Auxiliary <	🔬 Soft limit	Calibrati 1 0 No. 1 Yes	
✓ Settings	End load	Coordinate System Settings	Prefix Point name
	Friction comp.	Modify Clear Apply	Senter
	Speed scaling		
	10 filtering		Point name Add
	Ht Di config		Base coord. OFF
	DO config		Tool coord ON
			Ed avia a
	Contig import		Traiectory
			Import tool

Obrázek 4.3-5 Zobrazení souřadnicového systému vnější osy

## 1.3.5.3.3 Nastavení a zobrazení způsobu instalace robota

Výchozím režimem instalace robota je horizontální instalace. Při změně režimu instalace robota je třeba na této stránce včas nastavit aktuální režim instalace robota, aby byl zajištěn normální provoz robota.

Uživatel klikne na záložku "Pevná instalace" v oblasti 3D virtuálního zobrazení robota, čímž vstoupí na stránku nastavení režimu pevné instalace robota, vybere možnost "Montáž", "Překlopná montáž" nebo "Boční montáž" a kliknutím na tlačítko "Použít" dokončí nastavení režimu instalace robota.



#### Obrázek 4.3-6 Pevná instalace

S ohledem na flexibilnější a bohatší scénáře nasazení robotů poskytujeme funkci instalace M ee. Kliknutím na záložku "360-stupňová instalace M ee" v oblasti 3D virtuálního zobrazení robota vstoupí uživatelé na stránku nastavení režimu instalace robota M ee. Ručně nastavte úhly "náklonu základny" a "natočení základny" a 3D model podle toho zobrazí efekt instalace. Po úpravě kliknutím na tlačítko "Apply" (Použít) dokončete nastavení způsobu instalace robota.

a	=	🛞   🕞 🔳 🕕 Stopped toolcoord0 wobj0 exaxis0 0 🖢 🔺 🕅	8
🏟 Initialize		Operation& Status 360° Free Mounting Fixed Mounting	
🗐 Teaching	<		
- <b>/</b> → Status	<		
88 Auxiliary	<		Base tilt
🗙 Settings			( <b>1</b> 45*
		•	
			•
			•
			(45")
		Base rotation	
			Apply

Obrázek 4.3-7 Instalace 360 stupňů M ee

**Důležité:** Po dokončení instalace robota je nutné správně nastavit způsob instalace robota, jinak to ovlivní používání funkce tažení robota a funkce detekce kolizí.

## 1.3.5.4 Nastavení robota

## 1.3.5.4.1 Souřadnice nástroje

V nabídce "Robot Settings" (Nastavení robota) v části "Initial Settings" (Počáteční nastavení) klikněte na položku "Tool Coordinates" (Souřadnice nástroje), čímž vstoupíte do rozhraní souřadnic nástroje. Souřadnice nástroje mohou realizovat úpravu, vymazání a použití souřadnic nástroje. V rozevíracím seznamu souřadnicového systému nástroje je 15 čísel. Po výběru odpovídajícího souřadnicového systému (název souřadnicového systému lze přizpůsobit) se níže zobrazí odpovídající hodnota souřadnic, typ nástroje a pozice instalace (zobrazí se pouze u nástroje typu senzor), po výběru určitého souřadnicového systému klikněte na tlačítko "Použít" a aktuálně používaný souřadnicový systém nástroje se změní na vybrané souřadnice, jak je znázorněno na obrázku3.4-1.

Tool coord	inate sy	stem se	ttings	
Current tool	coordin	ate syste	m	
Coordinate name	system	toolo	coord7	•
X 0.000	Y	0.000	Z	200.000
RX 0.000	RY	0.000	RZ	0.000
Tool Type:	0		0: too	l, 1: sensor
Installati	0		0: en	d, 1: external
Coordinate	System	Settings		
Modify		Clear		Apply

Obrázek 4.4-1 Nastavení souřadnic nástroje

Kliknutím na tlačítko "Modify" obnovíte souřadnicový systém nástroje čísla podle výzvy. Metody kalibrace nástrojů se dělí na čtyřbodovou a šestibodovou metodu. Čtyřbodová metoda kalibruje pouze nástroj TCP, tedy polohu středového bodu nástroje. Jeho poloha je ve výchozím nastavení shodná s koncovou polohou. Šestibodová metoda přidává ke čtyřbodové metodě dva body., která se používá ke kalibraci polohy nástroje, zde si jako příklad pro vysvětlení vezmeme šestibodovou metodu.



Obrázek 4.4-2 Nastavení souřadnic nástroje

Vyberte pevný bod v prostoru robota, přesuňte nástroj do pevného bodu ve třech různých polohách a postupně nastavte 1-3 body. Jak je znázorněno v levém horním rohu obrázku 4.4-3. Přesuňte nástroj vertikálně k pevnému bodu nastavení bodu 4, jak je znázorněno v pravém horním rohu obrázku 4.4-3. Zachovejte nezměněnou polohu, použijte k pohybu základní souřadnice, posuňte se o určitou vzdálenost v horizontálním směru a nastavte bod 5, který je kladným směrem osy X nastaveného souřadnicového systému nástroje. Vraťte se do pevného bodu, posuňte se o určitou vzdálenost ve svislém směru a nastavte bod 6. Tento směr je kladným směrem osy Z souřadného systému nástroje a kladný směr osy Y souřadného systému nástroje je určen pravidlem pravé ruky. Kliknutím na tlačítko Vypočítat vypočítáme polohu nástroje. Pokud ji potřebujete vynulovat, klepněte na tlačítko Zrušit a stisknutím tlačítka Upravit znovu vytvořte souřadný systém nástroje.



Obrázek 4.4-3 Schéma šestibodové metody

Po dokončení posledního kroku se kliknutím na tlačítko "Finish" vraťte do rozhraní souřadnic nástroje a kliknutím na tlačítko "Save" uložte právě vytvořený souřadnicový systém nástroje.

# Důležité:

- 1. Po instalaci nástroje na konec je třeba kalibrovat a použít souřadnicový systém nástroje, jinak poloha a poloha středového bodu nástroje nebudou odpovídat očekávaným hodnotám, když robot provede příkaz k pohybu.
- 2. Souřadnicový systém nástroje obecně používá toolcoord1~toolcoord14 a toolcoord0 se používá k označení, že polohový střed nástroje TCP je ve středu koncové příruby. Při kalibraci souřadnicového systému nástroje je nutné nejprve použít souřadnicový systém nástroje toolcoord0 a poté vybrat další souřadnicové systémy nástroje pro kalibraci a použití.

# 1.3.5.4.2 Souřadnice externího nástroje

V nabídce "Robot Settings" (Nastavení robota) v části "Initial Settings" (Počáteční nastavení) klikněte na položku "External Tool Coordinate System" (Externí souřadnicový systém nástroje), čímž vstoupíte do rozhraní externího souřadnicového systému nástroje.

Úpravu, vymazání a použití externích souřadnic nástroje lze provést v rozhraní pro nastavení externího souřadnicového systému nástroje.

V rozevíracím seznamu externího souřadnicového systému nástroje je 15 čísel, M om etoolcoord0~etoolcoord14, po výběru příslušného souřadnicového systému se níže zobrazí odpovídající hodnota souřadnic, po výběru souřadnicového systému klikněte na tlačítko "Apply", aktuálně používaný souřadnicový systém nástroje se změní na vybrané souřadnice, jak je znázorněno na obrázku 4.4-4.

Coo	rdinate s	ystem r	name	etoolcoo	ord0
EX	0.000	EY	0.000	EZ	0.000
ERX	0.000	ERY	0.000	ERZ	0.000
тх	0.000	TY	0.000	TZ	0.000
TRX	0.000	TRY	0.000	TRZ	0.000

#### External tool coordinate system settings

# Obrázek 4.4-4

Kliknutím na tlačítko "Modify" (Upravit) přenastavte souřadnicový systém nástroje na číslo podle výzvy, jak je znázorněno na obrázku 4.4-5.



Obrázek 4.4-5 Schéma šestibodové metody

# 1. Tříbodová metoda pro určení vnějšího TCP

1. Nastavení bodu 1Typ měřeného nástroje se přesune na externí bod TCP, klikněte na tlačítko

Nastavení bodu 1; 2.

Nastavte bod 2Přesuňte bod 1 o určitou vzdálenost M podél osy X externího souřadnicového systému TCF a klikněte na tlačítko t

3.

Nastavte bod 3Go zpět na bod 1, přesuňte bod M om 1 podél osy Z vnějšího souřadnicového systému TCF na určitou vzdálenost,

4. VypočítatKliknutím na tlačítko vypočítat získáte externí TCF;

# 2. Šestibodová metoda pro stanovení TCF nástroje

1.

Nastavení bodů 1-4Výběr pevného bodu v prostoru robota, přesunutí nástroje do vybraného bodu M om čtyř různých úhlů a nastavení p

- Nastavení bodu 5Přejděte zpět do pevného bodu a přesuňte se o určitou vzdálenost podél osy X souřadného systému TCF nástroje a klikněte na tlačítko Nastavit bod 5;
- 3. Nastavení bodu 6Přejděte zpět do pevného bodu a přesuňte se o určitou vzdálenost podél osy Y souřadnicového systému TCF nástroje a klikněte na tlačítko nastavení bodu 6;
- 4. VypočítatKliknutím na tlačítko vypočítat získáte nástroj TCF;

Pokud potřebujete obnovit nastavení, klikněte na tlačítko Zrušit a vraťte se ke kroku vytvoření nového souřadnicového systému nástroje.

Po dokončení posledního kroku se kliknutím na tlačítko "Finish" vraťte do rozhraní souřadnic nástroje a kliknutím na tlačítko "Save" uložte právě vytvořený souřadnicový systém nástroje.

# Důležité:

- Použití externích nástrojů musí být kalibrováno a aplikováno na externí souřadnicový systém nástroje, jinak poloha a poloha středového bodu nástroje při provádění pohybových příkazů robotem nebude odpovídat očekávaným hodnotám.
- 2. Souřadnicový systém vnějšího nástroje obecně používá etoolcoord1~etoolcoord14 a použití etoolcoord0 znamená,

# <u>M, verze 1.0.0</u>

že středová poloha vnějšího nástroje TCP je ve středu koncové příruby. Při kalibraci nástroje

souřadnicový systém, je třeba nejprve použít souřadnicový systém nástroje etoolcoord0 a poté zvolit kalibraci ostatních souřadnicových systémů nástroje.

## 1.3.5.4.3 Souřadnice obrobku

V nabídce "Robot Settings" (Nastavení robota) v části "Initial Settings" (Počáteční nastavení) klikněte na položku "Workpiece Coordinates" (Souřadnice obrobku), čímž vstoupíte do rozhraní souřadnic obrobku. Souřadnice obrobku mohou realizovat úpravu, vymazání a použití souřadnic obrobku. V rozevíracím seznamu souřadnicového systému obrobku je 15 čísel, vyberte odpovídající souřadnicový systém (wobjcoord0~ wobjcoord14) a poté se v níže uvedeném okně "Souřadnice souřadnicového systému" zobrazí odpovídající hodnota souřadnice. Po výběru určitého souřadnicového systému klikněte na tlačítko "Použít" a aktuálně používaný souřadnicový systém obrobku se změní na vybrané souřadnice, jak ukazuje obrázek3.4-6 na obrázku.

porc	linate sy	stem r	name	wobjcod	ord0
X	000.0	Y	0.000	Z	0.000
x	0.000	RY	0.000	RZ	0.000
×	0.000	RY	0.000	RZ	0.000

Obrázek 4.4-6 Nastavení souřadnic obrobku

Souřadnicový systém obrobku je obvykle kalibrován na základě nástroje a souřadnicový systém obrobku je třeba vytvořit na základě vytvořeného souřadnicového systému nástroje. Kliknutím na tlačítko "Modify" (Upravit) obnovte souřadnicový systém obrobku čísla podle výzvy. Upevněte obrobek a zvolte metodu kalibrace "počátek osy-X-Z-" nebo "počátek osy-X-XY+rovina". Výběr prvních dvou bodů obou metod kalibrace je stejný, třetí bod se liší. Jedna metoda spočívá v kalibraci ve směru Z souřadného systému obrobku a druhá metoda spočívá v kalibraci bodu v rovině XY+, stačí provést kalibraci podle obrázku. Kliknutím na tlačítko Vypočítat vypočítáme polohu obrobku. Pokud ji potřebujete vynulovat, klepněte na tlačítko Zrušit a stisknutím tlačítka Upravit znovu vytvořte souřadnicový systém obrobku.





Po dokončení posledního kroku se kliknutím na tlačítko "Finish" vraťte do souřadnicového rozhraní obrobku a kliknutím na tlačítko "Save" uložte právě vytvořený souřadnicový systém obrobku.

#### Důležité:

- 1. Souřadnicový systém obrobku je kalibrován na základě nástroje a souřadnicový systém obrobku je třeba stanovit na základě stanoveného souřadnicového systému nástroje.
- 2. Souřadnicový systém obrobku obecně používá wobjcoord1~wobjcoord14 a wobjcoord0 se používá k označení, že počátek souřadnicového systému obrobku je v počátku základních souřadnic. Při kalibraci souřadnicového systému obrobku je nutné nejprve použít souřadnicový systém obrobku wobjcoord0 a poté vybrat další souřadnicové systémy obrobku pro kalibraci a použití.

## 1.3.5.4.4 Rozšířené osové souřadnice

V nabídce "Robot Settings" (Nastavení robota) v části "Initial Settings" (Počáteční nastavení) klikněte na položku "Extended Axis Coordinate System" (Rozšířený osový souřadnicový systém), čímž vstoupíte do rozhraní rozšířeného osového souřadnicového systému. V rozhraní pro nastavení rozšířeného osového souřadnicového systému lze realizovat úpravy, vymazání a použití rozšířených osových souřadnic.

V rozevíracím seznamu rozšířeného souřadnicového systému osy je 5 čísel, M om eaxis0~eaxis4, po výběru příslušného souřadnicového systému se níže zobrazí odpovídající hodnota souřadnice, po výběru souřadnicového systému klikněte na tlačítko "Použít", aktuálně používané rozšířené souřadnice osy Systém se stane vybranými souřadnicemi, jak je znázorněno na obrázku 4.4-8.

Coo	rdinate :	syste	em I	name e	exaxis0	•
х	0.000		Y	0.000	Z	0.000
RX	0.000		RY	0.000	RZ	0.000
Ext.	axis:	0				
Calil	brati	0			0: No	, 1: Yes

#### Obrázek 4.4-8 Rozšířené osové souřadnice

Kliknutím na tlačítko "Modify" (Upravit) přenastavíte rozšířený souřadnicový systém osy čísla podle výzvy, jak je znázorněno na obrázku 4.4-9. Před kalibrací vymažte rozšířený souřadnicový systém osy, který má být kalibrován, a použijte tento rozšířený souřadnicový systém osy. Podívejme se nejprve na první rozšiřující osový systém - metodu kalibrace lineárního vedení. Vybereme číslo rozšiřující osy, získáme informace, abychom získali informace o ovladači příslušné rozšiřující osy, a na základě těchto informací můžeme konfigurovat parametry. Po konfiguraci nastavíme parametry DH a schéma lineárního vedení je ve výchozím nastavení 0. Nastavíme polohu robota vzhledem k rozšiřující ose a lineární vedení je na rozšiřující ose. Pokud nechcete provádět kalibraci, stačí kliknout na tlačítko Uložit. V tuto chvíli se expanzní osa může pohybovat pouze asynchronně.

Degree of	freedom 1		
Ext. axis	1	•	
			Pick up info
Set zero	<i>1</i> .		Para. config
DH para. c	config.		
d1: 0	mm	a1: 0	mm
			Apply
Relative e	extended axi	s On exte	ension ax •
1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1			



Pokud se potřebujete pohybovat synchronně s robotem, kliknutím na Kalibrovat vstoupíte do kalibračního rozhraní. V nulovém bodě prodloužené osy klikněte na Eaxis v operační oblasti, abyste prodlouženou osu aktivovali, a vyrovnejte koncový střed robota (pomocí koncového bodu nástroje v souřadném systému nástroje aplikace) se dvěma různými polohami. Upevněte bod na kvazirozšířené ose, nastavte bod 1, resp. bod 2. Odstraňte povolení a posuňte prodlužovací osu o určitou vzdálenost. Po povolení také vyrovnejte středový bod konce robota s předchozím pevným bodem a nastavte bod 3. Odstraňte povolení, posuňte rozšiřovací osu do nulového bodu a povolte rozšiřovací osu. Přesuňte středový bod konce robota do pevného bodu a vertikálně do prostoru směrem nahoru, nastavte bod 4, vypočítejte souřadný systém a uložte jej.



Obrázek 4.4-10 Kalibrace lineárního vedení

Dále se podíváme na druhé schéma rozšiřující osy - metodu kalibrace polohovacího zařízení. Polohovač se skládá ze dvou rozšiřujících os. Vyberte číslo rozšiřující osy a získejte informace pro získání informací o ovladači příslušné rozšiřující osy. Podle těchto informací můžeme nastavit parametry. Po konfiguraci nastavte parametry DH, změřte parametry DH polohovadla podle schématu a zadejte je do vstupního pole. Nastavte polohu robota vzhledem k prodlužovací ose a polohovadlo je mimo prodlužovací osu. Pokud nechcete provádět kalibraci, stačí kliknout na tlačítko Uložit. V tuto chvíli se může rozšiřující osa pohybovat pouze asynchronně.

Extension axis scheme	e 1 - Positioner 🔻
Degree of freedom 1	
Ext. axis 1	¥
	Pick up info.
Set zero	Para. config.
Degree of freedom 2	
Ext. axis 1	T
	Pick up info.
Set zero	Para. config.
DH para. config.	
	Z.
*	1
	128.5
	3

Obrázek 4.4-11 Konfigurace polohovacího zařízení

Pokud se potřebujete pohybovat synchronně s robotem, kliknutím na Kalibrovat vstoupíte do kalibračního rozhraní. V nulovém bodě rozšířené osy klikněte na Eaxis v operační oblasti, abyste aktivovali rozšířenou osu, vytvořte souřadnicový systém na polohovacím zařízení, vyberte bod a zadejte hodnotu bodu pod souřadnicovým systémem. Carl polohuje, například vyberte bod v kladném směru Y a změřte Y na 100 mm, poté zadejte hodnotu podle obrázku, klikněte na referenční bod a referenční bod lze nastavit. Následující čtyři kalibrační body musí zarovnat střed konce robota (koncový bod nástroje v souřadném systému nástroje aplikace) s tímto referenčním bodem.



Obrázek 4.4-12 Konfigurace referenčního bodu polohovadla

Vyrovnejte střed konce robota (pomocí koncového bodu nástroje v souřadnicovém systému nástroje aplikace) na referenční bod, nastavte bod 1, klikněte na osu v operační oblasti, aby se obě osy na krátkou vzdálenost rozběhly, vyrovnejte střed konce robota na referenční bod a nastavte bod 2, pokračujte v pohybu obou os, střed konce robota se vyrovná s referenčním bodem, nastavte bod 3 a nakonec pokračujte v pohybu obou os, vyrovnejte střed konce robota s referenčním bodem, nastavte bod 4, klikněte na tlačítko Vypočítat a získejte souřadnice Kliknutím na tlačítko Uložit aplikujte výsledek.



Obrázek 4.4-13 Kalibrace polohovadla

Dále se podíváme na třetí rozšířené řešení osy - metodu kalibrace jednoosého polohovadla. Polohovač se skládá z rotujícího výsuvného hřídele. Vyberte číslo prodlužovacího hřídele a získejte informace pro získání informací o ovladači příslušného prodlužovacího hřídele. Podle těchto informací můžeme nastavit parametry. Parametr DH je nastaven na 0. Nastavíme polohu robota vzhledem k prodlužovací ose a polohovadlo je mimo prodlužovací osu. Pokud nechcete provádět kalibraci, stačí kliknout na tlačítko Uložit. V tuto chvíli se může rozšiřující osa pohybovat pouze asynchronně.

Degree of freedom 1 Ext. axis 1 • Pick Set zero Para	
Ext. axis 1  Pick Set zero Para	
Pick Set zero Para	
Set zero Para	c up info.
	a. config.
DH para. config.	
d1: 0 mm a1: 0	mm
	Apply
Relative extended axis position of robot On extension	ion ax 🔻
	Apply
1 <sup>2</sup> 2	

Obrázek 4.4-14 Konfigurace polohovacího zařízení pro jednu osu

Pokud se potřebujete pohybovat synchronně s robotem, kliknutím na Kalibrovat vstoupíte do kalibračního rozhraní. V nulovém bodě rozšířené osy klikněte na Eaxis v operační oblasti, abyste aktivovali rozšířenou osu, vytvořte souřadnicový systém na polohovacím zařízení, vyberte bod a zadejte hodnotu bodu pod souřadnicovým systémem. Carl pose, klikněte na "Reference Point" (Referenční bod) a nastavte referenční bod. Následující čtyři kalibrační body musí zarovnat střed konce robota (koncový bod nástroje v souřadném systému nástroje aplikace) s tímto referenčním bodem. Vyrovnejte střed konce robota (pomocí koncového bodu nástroje v souřadnicovém systému nástroje aplikace) s referenčním bodem, nastavte bod 1, kliknutím na "Eaxis" v operační oblasti rozběhněte osu otáčení na krátkou vzdálenost, vyrovnejte střed konce robota s referenčním bodem a nastavte bod 2. Poté klikněte na "Eaxis" v operační oblasti a nastavte bod 2. Pokračujte v pohybu osy otáčení, vyrovnejte střed konce robota s referenčním bodem, nastavte bod 3 a nakonec pokračujte v pohybu osy otáčení, vyrovnejte střed konce robota s referenčním bodem, nastavte bod 4, klikněte na tlačítko Vypočítat a získejte výsledek souřadnicového systému, kliknutím na tlačítko Uložit použijte.



Obrázek 4.4-15 Kalibrace polohovadla jedné osy

#### Důležité:

- 1. Souřadnicový systém rozšířené osy je kalibrován na základě nástroje a souřadnicový systém rozšířené osy je třeba vytvořit na základě vytvořeného souřadnicového systému nástroje.
- 2. Systém rozšířených os obecně používá exaxis1~exaxis4 a exaxis0 se používá pro souřadnicový systém bez rozšířených os. Při kalibraci rozšířeného osového souřadnicového systému musí být nejprve aplikován rozšířený osový souřadnicový systém na exaxis0 a poté by měly být vybrány další rozšířené osové souřadnicové systémy pro kalibraci a aplikaci.

# 1.3.5.4.5 Úroveň kolize

V nabídce "Robot Settings" (Nastavení robota) v části "Initial Settings" (Počáteční nastavení) klikněte na položku "Collision Level" (Úroveň kolize), čímž vstoupíte do rozhraní úrovně kolize.

Úroveň kolize je rozdělena na jednu až deset úrovní, přičemž detekce jedné až tří úrovní je citlivější a robot musí jet doporučenou rychlostí. Současně si můžete zvolit vlastní procentuální nastavení a 100 % odpovídá desáté úrovni. Strategie kolize může nastavit způsob zpracování robota po kolizi, který se dělí na zastavení chyby a nepřetržitý pohyb, a uživatel jej může nastavit podle konkrétních požadavků na použití. Jako například obrázek 4.4-16.

Collision class	Standard grade •	
J1 1(16N: •	Standard grade	1(16N) •
J4 1(16N) •	Custom Percent	1(16N) •
Level 1 recommer 3 30%	nded speed 10%, Level	2 20%, Lev
Level 1 recommer 3 30%	nded speed 10%, Level	2 20%, Lev Apply
Level 1 recommer 3 30% Collision strates	nded speed 10%, Level	2 20%, Lev Apply

Obrázek 4.4-16 Schéma kolizní úrovně

# 1.3.5.4.6 Měkký limit

V nabídce "Robot Settings" (Nastavení robota) v části "Initial Settings" (Počáteční nastavení) klikněte na položku "Soft Limits" (Měkké limity), čímž vstoupíte do rozhraní měkkých limitů.

V tahu robota se mohou nacházet další zařízení a mezní úhel může jemně omezit robota tak, aby jeho pohyb nepřekročil určitou souřadnicovou hodnotu a zabránil kolizi robota. Spuštění měkkého limitu pro zastavení robota je automaticky spuštěno robotem a není zde žádná brzdná dráha.

Správci mohou použít výchozí hodnoty nebo zadat úhlové hodnoty. Zadáním hodnoty úhlu omezíte kladné, resp. záporné úhly kloubů robota. Pokud zadaná hodnota překročí měkký mezní úhel kloubů robota uvedený v tabulce 1.1-1 Základní parametry robotahodnota, bude mezní úhel upraven na maximální hodnotu, kterou lze nastavit. Když robot ohlásí, že příkaz překračuje mezní úhel, musí přejít do režimu přetahování a přetáhnout klouby robota do mezního úhlu. Rozhraní je znázorněno na obrázku 3.4-17.



Obrázek 4.4-17 Schematické schéma oMobot limit

#### 1.3.5.4.7 Koncové zatížení

V nabídce "Robot Settings" (Nastavení robota) v části "Initial Settings" (Počáteční nastavení) klikněte na položku "End Load" (Koncové zatížení), čímž vstoupíte do rozhraní koncového zatížení.

Uživatel může nastavit odpovídající parametry podle parametrů použitého nástroje. Hmotnost břemene je 0~5 kg a rozsah souřadnic středu hmotnosti je 0~1000, jak je znázorněno na obrázku 3.4-18.

Load weigh	t 2.00	0		kg
				Apply
Load centro	oid coor	dinate set	ting	
and the second	1.50	12/12/2/2	1 4	

Obrázek 4.4-18 Schéma nastavení zátěže

**Důležité:** Po instalaci zátěže na konec robota je třeba správně nastavit hmotnost koncové zátěže a souřadnice středu hmotnosti, jinak to ovlivní funkci odporu robota a použití funkce detekce kolize.

Pokud si uživatel není jistý kvalitou nástroje nebo středem hmotnosti, lze údaje o nástroji určit pomocí funkce identifikace zatížení.

Před měřením se ujistěte, že je zátěž nainstalována. Kliknutím na tlačítko "Tool Data Measurement" (Měření dat nástroje) vstupte do rozhraní pro testování pohybu zátěže.



Obrázek 4.4-19 Nastavení kloubu pro identifikaci zatížení

Klikněte na tlačítko "Load Identification Start" a proveďte test. V případě nouze pohyb včas zastavte.





Po skončení cvičení klikněte na tlačítko "Získat výsledek identifikace", čímž získáte vypočtená data nástroje a zobrazíte je na stránce. Pokud je chcete použít na údaje o zatížení, klikněte na tlačítko Apply (Použít).



Obrázek 4.4-21 Výsledky identifikace zatížení

#### 1.3.5.4.8 Miction kompenzace

V liště nabídek "Robot Settings" (Nastavení robota) v části "Initial Settings" (Počáteční nastavení) klikněte na položku "Miction Compensation" (Kompenzace diktátu) a vstupte do rozhraní pro nastavení kompenzace diktátu.

**Koeficient kompenzace m** ikceScénář použití pro kompenzaci m ikce je pouze v režimu přetahování. Koeficient kompenzace Miction lze nastavit M om 0 až 1. Čím vyšší je hodnota, tím větší je kompenzační síla při tažení. Koeficient kompenzace M iction je třeba nastavit pro každou osu zvlášť podle různých způsobů instalace.

**Přepínač kompenzace mikceUživatelé** mohou zapnout nebo vypnout kompenzaci mikce podle aktuálního robota a zvyklostí používání.

-perioditi	on coemcie	ent	
Horizo	ontal ins 🔹		
J2	0	J3	0
J5	0	J6	0
			Apply
npensati	on	pen	,
	Horizo	Horizontal ins  U2 U U U U U U U U U U U U U U U U U U	Horizontal ins  U2 0 J3 J5 0 J6 Inpensation

Obrázek 4.4-22 Nastavení kompenzace m iction Compensation

## Důležité:

Funkci M iction compensation robota je třeba používat s opatrností. Podle aktuálnísituace je přiměřená kompenzace.

# 1.3.5.4.9 Škálování rychlosti

V nabídce "Robot Settings" (Nastavení robota) v části "Initial Settings" (Počáteční nastavení) klikněte na položku "Speed Scaling Settings" (Nastavení škálování rychlosti), čímž vstoupíte do rozhraní pro nastavení škálování rychlosti.

Tato funkce slouží k nastavení rychlosti chodu robota v ručním/automatickém režimu. Pokud je aktuální režim chodu automatický, je nastavená rychlost automatickou rychlostí chodu robota. Pokud je aktuální režim chodu manuální, nastavená rychlost je manuální rychlost chodu robota. Nastavená hodnota je procento standardní rychlosti robota. Pokud je nastavena na 100, znamená to 100 % standardní rychlosti (standardní rychlost naleznete v tabulce 1.1-1 Základní parametry robota).

Speed scaling	settings	
Speed scaling	10	%
		Apply

## Obrázek 4.4-23 Nastavení měřítka rychlosti

Po úspěšném nastavení rychlosti se příslušný stavový řádek rychlosti změní na nastavenou hodnotu a rozsah nastavení hodnoty rychlosti je 0~100.

## 1.3.5.5 Vstupy/výstupy řídicího boxu

# 1.3.5.5.1 Nastavení I/O

Kliknutím na tlačítko "IO" v operační oblasti na levé straně 3D modelu vstoupíte do rozhraní pro nastavení IO, jak je znázorněno na obrázku 4.5-1. V tomto rozhraní lze realizovat digitální výstup, analogový výstup (0-10v) a digitální výstup koncového nástroje v ovládacím poli robota. Výstup, analogový výstup (0-10v) pro ruční ovládání:

Joir	nt Base	Tool	Wobj	Move	:fi.
Eax	is IO	TPD	FT	RCM	
Ctrll	Вох	2			
DO	DO0	۲			OFF
AO	Aout0	•		%	Set
End	Eff			1.5	
DO	DO0	•			OFF
00	·				

Obrázek 4.5-1 Rozhraní pro nastavení I/O

- Operace DO: vyberte číslo portu, pokud je DO na nízké úrovni, zobrazí se pravé tlačítko operace ON, kliknutím na tlačítko nastavte DO na vysokou úroveň.
- Operace AO: Hodnota je v procentech, nastavení 100 znamená nastavení portu AO na 10V.

## 1.3.5.5.2 Zobrazení stavu I/O

V oblasti zobrazení stavu na pravé straně 3D modelu se zobrazí aktuální stav IO. V případě digitálního vstupu a digitálního výstupu, pokud je úroveň portu vysoká, bude bod zobrazen zeleně, a pokud je nízká, bude zobrazen bíle; analogový vstup a analogový výstup Hodnota zobrazení je 0-100, přičemž 100 znamená 10V.

) DI0	() DI1		
O DO0	⊖ D01		
			EndEf
Ain0 :	0%	Q Ain1 :	0%
Aout0 :	0%	Q Aout1 :	0%
1014	0.010	0,010	U CII
CIU	CIA	O CI2	C CI3
O DI4	O DIS	D16	ODI7
O DI0	O DI1	O DI2	O DI3
○ CO4	_ CO5	○ CO6	○ C07
⊙ CO0	⊙ CO1	⊙ CO2	○ CO3
O DO4	○ DO5	) DO6	⊖ D07
) DO0	() D01	O DO2	O DO3
			CtrlBox

Obrázek 4.5-2 Rozhraní stavového displeje

## 1.3.5.5.3 Filtrování I/O

Klikněte na položku "Robot Settings" (Nastavení robota) v části "Initial Settings" (Počáteční nastavení) v levém panelu nabídek a kliknutím na podnabídku "IO Filter" (Filtr IO) vstupte do rozhraní pro nastavení času filtru IO. Rozhraní pro nastavení času filtru zahrnuje: čas filtrování DI řídicího boxu, čas filtrování DI koncové desky, čas filtrování AIO řídicího boxu, čas filtrování AI1 řídicího boxu, čas filtrování AI0 koncové desky, jak je znázorněno na obrázku3.5-3. Uživatelé mohou nastavit příslušné parametry podle svých potřeb, stačí kliknout na příslušné tlačítko nastavení.



Obrázek 4.5-3 Rozhraní filtru

Důležité: Časový rozsah filtru I/O je [0~200], jednotka je ms.

## 1.3.5.5.4 Konfigurace I/O

Klikněte na položku "Robot Settings" (Nastavení robota) v části "Initial Settings" (Počáteční nastavení) na levém panelu nabídky a kliknutím na podnabídky "DI Configuration" (Konfigurace DI) a "DO Configu- ration" (Konfigurace DO) vstupte do rozhraní pro konfiguraci DI a DO. Mezi nimi jsou konfigurovatelné ovládací pole CIO-CI7 a CO0-CO7 a terminály DI0 a DI1. Ve výrobě, když kolaborativní robot potřebuje připojit periferní zařízení nebo se náhle zastaví z důvodu poruchy nebo jiných faktorů, je třeba, aby vydal signál DO a realizoval zvukovou a světelnou výstražnou výzvu. Vstupní konfigurovatelné funkce jsou uvedeny v tabulce 4.5-1 Vstupní konfigurovatelné funkce řídicí skříňky.

Tabulka 4.5-1 Konfigurovatelné funkce vstupů ovládacího pole

Funkce č.	Název funkce
0	nic
1	Signál úspěchu oblouku
2	Signál pro přípravu svařovacího stroje
3	Detekce pásu
4	pozastavit
5	zotavení
6	start-up
7	zastavit
8	Pauza/pokračování
9	Spuštění/zastavení
10	Přepínač tahu pedálu
11	Přesun do místa původu
12	Ruční automatické přepínání
13	Zjištění polohy svařovacího drátu se podařilo
14	Přerušení pohybu
15	Spuštění hlavního programu
16	Spustit přetáčení
17	Potvrzení spuštění
18	Detekční signál laseru X
19	Detekční signál laseru Y

Výstupní konfigurovatelné funkce jsou uvedeny v tabulkách 4.5-2 a 4.5-3.

Tabulka 4.5-2 Konfigurovatelná	funkce výstupu	ovládací skříňky
Tubuna 1.5 2 Koningar ovatema	Turnee vystupu	oviduaci ski niky

Funkce č.	Název funkce
0	nic
1	hlášení chyb
2	motion
3	Spuštění a zastavení postřiku
4	Čištění stříkací pistole
5	Oblouk
6	aspirace
7	Přívod drátu dopředu
8	Reverzní podávání drátu
9	Vstupní port JOB 1
10	Vstupní port JOB 2
11	Vstupní port JOB 3
12	Spuštění a zastavení dopravníkového pásu
13	pozastavit
14	Dosažení původu práce
15	Vstup do rušivé zóny
16	Řízení polohy svařovacího drátu Start-Stop
17	Spuštění robota dokončeno
18	Spuštění a zastavení programu
19	Automatický manuální režim

Tabulka 4.5-3 Konfigurovatelná funkce vstupů svorek

Funkce č.	Název funkce
0	nic
1	Režim přetahování
2	Výukový bodový záznam
3	Ruční automatické přepínání
4	Spuštění/zastavení záznamu stopy TPD
5	pozastavit
6	zotavení
7	start-up
8	zastavit
9	Pauza/pokračování
10	Spuštění/zastavení

Mezi nimi je výchozí konfigurace ovládacího pole: CO0 je 1 - robot hlásí chybu a CO1 je 2 - robot je v pohybu.

Dico	onfiguration				Do co	onfiguration		
Cont	rol box input				C00	report errors	C01	motion
C10	nothing •	CI1	nothing	•	CO2	Spraying star	CO3	Spray gun cle
CI2	nothing •	CI3	nothing	•	CO4	nothing •	CO5	nothing
CI4	nothing •	CI5	nothing	•	CO6	nothing •	C07	nothing
CI6	nothing •	CI7	nothing	*				and the second second
CI0	Active at low   •	CI1	Apply Active at low I	•	C00	Active at high •	]C01	Active at high
C12	Active at low   •	CI3	Active at low I	•	CO2	Active at high •	CO3	Active at high
CI4	Active at low I •	CI5	Active at low I	•	CO4	Active at high •	CO5	Active at high
CI6	Active at low   •	CI7	Active at low I	•	CO6	Active at high •	C07	Active at high
			Apply					Apply

Obrázek 4.5-4 Konfigurace řídicí skříňky DI a DO

Výchozí konfigurace svorky DI: DI0 táhne výuku, DI1 výukový bod záznamu.





Po dokončení konfigurace můžete zobrazit odpovídající stav výstupu DO na stránce I/O ovládacího panelu pod příslušným stavem. (Poznámka: nakonfigurované DI a DO se nemohou zobrazit na stránce programování výuky).

Důležité: Nakonfigurované DI a DO je zakázáno používat při výuce programování.

## 1.3.5.6 Provoz robota

## 1.3.5.6.1 Výukový bodový záznam

Oblast ručního ovládání výuky slouží především k nastavení zkušebního souřadnicového systému v režimu výuky a k zobrazení hodnot úhlu a souřadnic každé osy robota v reálném čase a k pojmenování a uložení výukových bodů.

Při ukládání výukového bodu je souřadnicovým systémem výukového bodu souřadnicový systém aktuální aplikace robota. Rychlost a zrychlení výukového bodu lze nastavit nad operační oblastí. Nastavená hodnota je procento standardní rychlosti robota. Pokud je nastavena na 100, znamená to 100 % standardní rychlosti (vizTabulka 1.1-1 Základní parametry robota.

Pro výukový bod senzoru vyberte nástroj kalibrovaný typ senzoru, zadejte název bodu, klikněte na tlačítko Přidat a uložená poloha bodu je poloha rozpoznaná senzorem.



Obrázek 4.6-1 Schéma oblasti ručního ovládání

**Důležité: P ř i** prvním použití nastavte malou hodnotu rychlosti, například 30, abyste se seznámili s pohybem robota a předešli nehodám.

# 1.3.5.6.2 Společný běh

V části Joint operation (Kloubová operace) představuje 6 posuvníků uprostřed úhly příslušných os a pohyb kloubu je rozdělen na jednoosé a víceosé propojení.

**jednoosý** jogUživatel může ovládat pohyb robota pomocí kruhových tlačítek vlevo a vpravo, jak je znázorněno na obrázku3.6-2. V manuálním režimu a souřadném systému kloubů proveďte operaci otáčení kloubu robota. Když se robot zastaví mimo rozsah pohybu (měkký limit), můžete pro ruční ovládání použít jednoosý jogging, abyste robota vyvedli z polohy překročení. Jednoosý jogging je rychlejší a pohodlnější než ostatní operační režimy pro hrubé polohování a rozsáhlé pohyby.

Nastavte parametry "prahu pohybu dlouhým stisknutím" (maximální vzdálenost, kterou může robot uběhnout při dlouhém stisknutí tlačítka, a rozsah vstupních hodnot je 0 ~ 300), dlouhým stisknutím kulatého tlačítka ovládejte chod robota, pokud je tlačítko uvolněno během chodu robota, robot se okamžitě zastaví Pohyb, pokud budete tlačítko stále mačkat, aniž byste ho uvolnili, robot se rozběhne a přestane se pohybovat po dlouhém stisknutí hodnoty nastavené prahem pohybu. Víceosé propojeníUživatel může ovládat šest posuvníků uprostřed a nastavit tak odpovídající cílovou polohu robota, jak je znázorněno na obrázku3.6-3. Cílovou polohu lze určit pozorováním 3D virtuálního robota. Pokud nastavená poloha nesplňuje vaše očekávání, klikněte na tlačítko "Obnovit". Přimějete 3D virtuálního robota, aby se vrátil do původní polohy. Poté, co uživatel určí cílovou polohu, může kliknout na tlačítko "Použít" a fyzický robot provede odpovídající pohyby.



Obrázek 4.6-2 Schéma jednoosého joggingu a víceosého propojení

**Důležité:** U víceosého propojení nesmí být hodnota nastavení pátého kloubu j5 menší než 0,01 stupně. Pokud je očekávaná hodnota menší než 0,01 stupně, můžete ji nejprve nastavit na 0,011 stupně a poté doladit pátý kloub j5 pomocí jednoosého běhu.

# 1.3.5.6.3 Základna Jog

V základním souřadnicovém systému můžete ovládat robota pomocí kruhových tlačítek na levé a pravé straně, pohybovat se po přímce v osách X, Y a Z nebo se otáčet kolem os RX, RY a RZ a šest posuvných lišt uprostřed je příslušně znázorněno na odpovídajících souřadnicových osách Poloha a rozsah pohybu, jak je znázorněno na obrázku 3.6-3. Funkce základního běhu je podobná funkci jednoosého běhu ve společném pohybu.



Obrázek 4.6-3 Základní jog diagram

**Důležité:** Tlačítko lze kdykoli uvolnit, aby se robot zastavil. V případě potřeby robota zastavte stisknutím tlačítka nouzového zastavení.

# 1.3.5.6.4 Nástroj Jog

Vyberte souřadnicový systém nástroje, můžete ovládat robota pomocí kruhových tlačítek na levé a pravé straně, pohybovat se po přímce na osách X, Y a Z nebo se otáčet kolem os RX, RY a RZ a šest posuvníků uprostřed představuje polohy na příslušných souřadnicových osách. Poloha a rozsah pohybu, jak je znázorněno na obrázku3.6-4. Funkce Tool jogging je podobná jako u jednoosého joggingu ve funkci Joint motion.



Obrázek 4.6-4 Schematické schéma rozbíhání nástrojů

# 1.3.5.6.5 Wobj jog

Zvolte běh obrobku, robota můžete ovládat kruhovými tlačítky vlevo a vpravo. V souřadném systému obrobku se pohybujte podél os X, Y, Z v přímce nebo se otáčejte kolem RX, RY, RZ. Šest posuvníků uprostřed je v tomto pořadí Označuje polohu a rozsah pohybu na příslušné souřadnicové ose, jak je znázorněno na obrázku3.6-5. Funkce joggingu Wobj je podobná jako u jednoosého joggingu v režimu Joint motion.



Obrázek 4.6-5 Schéma běhu Wobj

## 1.3.5.6.6 Přesun

Vyberte možnost Move, můžete přímo zadat hodnotu kartézské souřadnice, klikněte na tlačítko "Calculate joint position" (Vypočítat polohu kloubu), poloha kloubu se zobrazí jako výsledek výpočtu, potvrďte, že nehrozí žádné nebezpečí, a kliknutím na tlačítko "Move to this point" (Přesunout do tohoto bodu) můžete ovládat robota, aby se přesunul do zadané kartézské polohy.

ax	uis IO	TPE	F	r R	СМ
00		linate p	ositio	on	
Х	0	mr	n RX	0	•
Y	0	mr	n RY	0	•
	1	-	- 07	0	
Z	0	mr	II RZ	Calcu	late joint
Z		tion	II RZ	Calcu	late joint
D D J T	0 nt posi	tion	J4	Calcu	late joint
oir J1	0 nt posi 0	tion °	J4 J5	Calcu O	late joint
Z Dir J1 J2 J3	0 nt posi 0 0	tion °	J4 J5 J6	0 Calcu 0 0	late joint

Obrázek 4.6-6 Schéma přesunu mobilních zařízení

**Důležité:** Pokud nelze dosáhnout dané pozice, nejprve zkontrolujte, zda pozice v kartézském prostoru nepřekračuje pracovní rozsah robota, a poté zkontrolujte, zda v procesu M om aktuální pozice k cílové pozici neexistuje singulární pozice, pokud existuje singulární pozice, upravte aktuální pozici nebo do procesu vložte novou pozici, abyste se vyhnuli singulárním pozicím.

## 1.3.5.6.7 Pohyb na ose Eaxis

Zvolte Eaxis to move, tato funkce je funkcí jog přídavné osy, je třeba ji nakonfigurovat pro konfiguraci přídavné osy, použijte tuto funkci jog k ovládání přídavné osy, podrobnosti o konfiguraci viz "Kapitola 4 Periferie stroje a robota - Periferie přídavné osy".".
Joint Base	Tool Wobj M	love
Eaxis IO	TPD FT R	CM
Ext. axis	1 •	
Running	100	%
Accelera	100	%
Max dist	50	(mm)(°)
Disable	Set zero	Enable
Stop	Reverse	Forward
Prefix		
Point name		Add

Obrázek 4.6-7 Schéma mobilního zařízení Eaxis

# 1.3.5.6.8 TPD (výukové programování)

Kroky funkce programování výuky (TPD) jsou následující:

- Krok1 záznam počáteční polohyVstupte do operační oblasti na levé straně 3D modelu a zaznamenejte aktuální polohu robota. V editačním poli nastavte název bodu, klikněte na tlačítko "Uložit", pokud se uložení podaří, zobrazí se výzva "Uložení bodu proběhlo úspěšně";
- Krok2 Konfigurace parametrů záznamu trajektorieKliknutím na položku TPD vstupte do položky funkce "TPD" a nakonfigurujte parametry záznamu trajektorie, nastavte název souboru trajektorie, typ pózy a periodu vzorkování, nakonfigurujte DI a DO a zaznamenejte odpovídající výstup spuštěním DI během procesu záznamu trajektorie TPD DO, jak ukazuje obrázek 3.6-8;

Joint Base Eaxis 10	Tool Wobj	M R	ove CM
tpd track re	cord		
Track na	+		
Posture	Joint posture	•	
Period	2	•	
Di config.	nothing	•	
Do config.	nothing	•	
TPD-State:	Not recording		
Config.	Start	416 - 210	Stop
Tpd track e	diting		
Track na		۲	Get points
Start			
	0		-0
End			
-	0		
Simulate			Finish
Track na		•	Delete

Obrázek 4.6-8 Dosavadní výsledky TPD

Krok3 Zkontrolujte režim robotaZkontrolujte, zda je režim robota v ručním režimu. Pokud ne, přepněte do
manuálního režimu. V manuálním režimu existují dva způsoby přepnutí do režimu výuky přetažením, jedním
je dlouhé stisknutí koncového tlačítka a druhým je režim přetažení rozhraní Doporučuje se přepnout robota M
om rozhraní do režimu výuky přetažením v záznamu TPD. Jak je znázorněno na obrázku 4.6-9;

Stopped tool1 wobj1 exaxis1 100 📿 🛦 👿 💋 😣

Obrázek 4.6-9 režim robota

### Důležité:

Při přepínání do režimu přetahování M om rozhraní nejprve ověřte, zda je správně nastaveno zatížení koncového nástroje a hmotnostní střed, a tlačítkem End potvrďte, zda je přetahování normální, a po potvrzení přepněte do režimu přetahování M om rozhraní.

#### že je správný.

- Krok4 zahájení nahráváníKliknutím na tlačítko "Start Recording" spusťte nahrávání stopy a přetáhněte robota k výuce činností. Kromě toho je v konfiguraci terminálu DI k dispozici konfigurační položka funkce "TPD recording start/stop". Konfigurací této funkce může uživatel spustit funkci "spuštění záznamu" stopy prostřednictvím externího signálu. Je třeba poznamenat, že chcete-li spustit záznam stopy prostřednictvím externího signálu, nejprve na této stránce nakonfigurujte informace o stopě TPD.
- Krok5 zastavení nahráváníPo dokončení výuky akce kliknutím na tlačítko "Zastavit nahrávání" zastavte nahrávání stopy a poté přetáhněte tlačítko přepínače výuky, aby robot ukončil režim výuky přetažením. Když se na učícím přívěsku objeví zpráva "stop track recording successful", znamená to, že nahrávání stopy proběhlo úspěšně. Stejně jako v kroku 4 můžete po konfiguraci funkce "TPD recording start/stop" spustit zastavení záznamu externím signálem.
- Krok6 učit programováníKlikněte na tlačítko Nový, vyberte prázdnou šablonu, klikněte pro vstup do
  položky programování funkce PTP, vyberte právě uložený počáteční polohový bod, klikněte na tlačítko
  "Přidat", po dokončení aplikace se v souboru programu zobrazí instrukce PTP; poté klikněte pro vstup do
  položky programování funkce TPD, vyberte právě zaznamenanou stopu, nastavte, zda je hladká, a škálování
  rychlosti, klikněte na tlačítko "Přidat", po dokončení aplikace se v souboru programu zobrazí instrukce
  MoveTPD, jak ukazuje obrázek 3.6-10;



Obrázek 4.6-10 Programování TPD

- Krok7 Opakování trajektoriePo úpravě výukového programu přepněte do režimu automatického provozu, kliknutím na ikonu "start running" v horní části rozhraní spusťte program a robot začne reprodukovat výukovou akci.
- Krok8 editace stopyV oblasti editace stopy TPD lze vizuálně zobrazit a upravit stopu a dosáhnout tak předběžné analýzy a zefektivnění stopy TPD. Vyberte příslušný bod pořízení stopy, poté se body stopy zaznamenané uživatelem zobrazí v trojrozměrném prostoru robota a uživatel pak může přetažením posuvníků "Start" a "End" simulovat a reprodukovat a upravovat počáteční a koncové body stopy.

Mazání souborů TPD a zpracování výjimek

 Odstranění souboru trajektorieKliknutím vstupte do položky funkce TPD, vyberte soubor trajektorie, který chcete odstranit, a klikněte na tlačítko "Odstranit trajektorii". Pokud je odstranění úspěšné, zobrazí se výzva k úspěšnému odstranění.

- Řešení výjimek
  - Překročení příkazových bodůSledování může zaznamenat až 20 000 bodů. Když počet překročí 20 000 bodů, řídicí jednotka již nebude zaznamenávat překročené body a vyšle výstrahu do učícího přívěsku, že počet příkazových bodů překročil limit. V tomto okamžiku je třeba kliknutím zastavit nahrávání;
  - Interval pokynů TPD je příliš velkýPokud výukový přívěsek hlásí chybu, že interval pokynů TPD je příliš velký, zkontrolujte, zda se robot před záznamem vrátil do výchozí polohy. Pokud se robot vrátí do výchozí polohy a stále hlásí chybu, že interval příkazu TPD je příliš velký, smažte aktuální stopu a nahrajte novou;
  - Pokud se během provozu TPD vyskytnou jiné abnormální stavy, je třeba provoz robota okamžitě zastavit pomocí učicího přívěsku nebo tlačítka nouzového zastavení a zkontrolovat příčinu.

Důležité: Při ovládání funkce TPD je třeba důsledně dodržovat příslušné pokyny na výukovém přívěsku.

#### 1.3.5.7 Výuková simulace

#### 1.3.5.7.1 Úvod

Kliknutím na příkaz vlevo přidáte uzel programu do stromu programů.Když program běží, je aktuálně prováděný uzel programu zvýrazněn šedě.V ručním režimu klikněte na první ikonu na pravé straně uzlu, aby robot provedl instrukci sám, a druhá ikona slouží k úpravě obsahu uzlu.



Obrázek 4.7-1 Stromové rozhraní programu

Kliknutím na tlačítko "" přepnete režimy a text výukového programu se přepne do stavu editace a oblast editace lze ve stavu editace rozbalovat a sbalovat.



Obrázek 4.7-2 Stav úprav výukového programu

Kliknutím na tlačítko "Rozbalit/skrýt obsah na pravé straně aktuálního programu" rozbalíte nebo skryjete místní výukové body a záložní obsah aktuálního programu. Po rozbalení obsahu vpravo klikněte na ikony "Dílčí výukový bod" a "Záloha aktuálního programu" pro zobrazení odpovídajícího obsahu.



Obrázek 4.7-3 Obsah na pravé straně aktuálního programu

#### 1.3.5.7.2 Panel nástrojů

Upravte strom programů pomocí panelu nástrojů v dolní části stromu programů.



**Poznámka:** nameOpen efektOtevřít soubor uživatelského programu



#### Poznámka:

### názevNové sestavení

effectVýběr šablony pro vytvoření nového programového souboru



# Poznámka:

#### nameImport

efektImportovat soubor do složky uživatelského programu



### Poznámka:

nameExport

effectExportovat soubory uživatelského programu do místního bodu.



# Poznámka:

názevUložit

efektUložit úpravy

souboru



## Poznámka:

nameUložit jako

efektPřejmenujte soubor a uložte jej do složky uživatelského programu nebo šablony programu.



#### Poznámka:

#### nameCopy

effectDuplikuje uzel a umožňuje jeho použití pro další operace (např.: vložení na jiné místo ve stromu programu).



#### Poznámka:

#### namePaste

effectUmožňuje vložit dříve vyříznuté nebo zkopírované uzly.



# Poznámka:

#### nameTo cut

effectOdřízne uzel a umožní jeho použití pro další operace (např.: vložení jinam ve stromu programu).



# Poznámka:

#### nameDelete

effectSmaže uzel M ze stromu programu.



#### Poznámka:

názevPřesunout nahoru

efektPřesun uzlu nahoru.



### 1.3.5.7.3 Programový příkaz

Levá strana slouží především k přidávání programových příkazů. Kliknutím na ikonu nad každým klíčovým slovem vstoupíte do podrobného rozhraní. Existují dvě hlavní operace pro přidávání programových příkazů do souboru. Jedním způsobem je otevřít příslušný příkaz a kliknutím na tlačítko Použít přidat příkaz do programu. , druhým způsobem je nejprve kliknout na tlačítko "Přidat", v tomto okamžiku se příkaz neuloží do souboru programu a je třeba znovu kliknout na tlačítko "Použít", aby se příkaz uložil do souboru. Druhý způsob se často vyskytuje v případě, že je vydáno více příkazů stejného typu. K tomuto typu příkazu přidáme tlačítko Přidat a zobrazíme obsah přidaného příkazu. Klepnutím na tlačítko Přidat přidáte příkaz a přidaný příkaz zobrazí všechny přidané příkazy. , klepnutím na tlačítko Použít uložíte přidatý příkaz do otevřeného souboru vpravo.

#### 1.3.5.7.4 Rozhraní příkazů logiky



# Obrázek 4.7-4 Rozhraní logických příkazů

# 1.3.5.7.4.1 Zatímco příkaz

Kliknutím na ikonu "While" vstoupíte do rozhraní pro úpravu příkazu While.

Do vstupního pole za tlačítkem While zadejte čekací podmínku, do vstupního pole za tlačítkem do zadejte akční příkaz během smyčky a klikněte na tlačítko Uložit. (Pro usnadnění práce můžete obsah příkazu do zadat libovolně a místo něj upravit jiné příkazy v programu).



Obrázek 4.7-4-1 Zatímco příkazové rozhraní

#### 1.3.5.7.4.2 pokud. . else

Klikněte na tlačítko "if. . else" vstoupíte do příkazu if. . . else rozhraní pro úpravu příkazů

Zadejte příkaz do vstupního pole vpravo a po úpravě klikněte na tlačítko "Přidat" a "Použít". (Tento pokyn vyžaduje určitý programátorský základ, pokud potřebujete pomoc, kontaktujte nás).

<i>ъ</i> ≡		<u>ن</u> ا ک	I		Stopped toolco	oord0 wobj0 exaxis	0   10			12 8
Initialize <	📹 🖪 🛎	⊥∎∎ ≯ (		felse >	Coperation&Stat	360" Free Mounting Fr	ixed Mount	ing		
🔲 Teaching 🗸	Logic commant~	Process1A.lua	example	f(Judgment Condition 1)then	Nove (m)		Robot P	ose		Œ
Program Teachi	00 7	1↔		Condition 1 is satisfied to execute the command			11 - 118	102 12 13	PT - 929-07	Joints
Graphical Program	While IT_Else	2⊷ SetWOb)Coord(1,464.500,820.000		Condition 2 is established to	2/3*2		J4 :-107.	B64 J6:-8	at. 160.1i	. 39.719
Manage Teaching	Goto Wat	3↔ SetToolCoord(2,-2.500,0.000,644		else			X 1990.3	00 V 14	71 509 2	TCP
4. Status <	II Đ	d↔ MoveJ( 0.670, .65.310, 134.370, -1	Program			-	RX:-151.	095 R(Y) 7	2.646 RZ	1-86.909
🔠 Auxillary <	Pause Dolle	5++ SPLCSetDO(1,1)			-138.897		Tota			FT
🛠 Settings	VW	0++ SPLCSetAO(0,10)			-79.428		Fx : 0.000	Fy:0 Ty:0	000 Fz	1.0.000
		7↔ SPLCSetAO(1,0)		-	75	2	Act_State	<u>نا</u>		
		B⊷ WaitDI (1,1,0,2)		Add	-107.864		Murr + D	000		Line-Num
		B++ SPLCSetDO(1,0)	Added Commands		-91.891					Chillon
		10 SPLCSetDO(7, 1)			39.719		DOG	DOT	002	DO3
		t1⊷ WaitMs(20)					000 CO4	C01 C05	CO2	001
		12 SPLCSetDO(7, 0)		Apply	Add		Dia	Dis	012	Dia
		13⊷ SetDO(11,1,0,0)		Sensor	~		CI0 CI4	CH	CI2 CIE	CID CID

Obrázek 4.7-4-2 pokud . . else příkazové rozhraní

### 1.3.5.7.4.3 Příkaz Goto

Kliknutím na tlačítko "Goto" vstoupíte do rozhraní pro úpravu příkazu Goto.

Instrukce Goto je instrukce skoku, zadejte příkaz do vstupního pole vpravo a po úpravě klikněte na tlačítko "Přidat" a "Použít". (Tato instrukce vyžaduje určitý programátorský základ, pokud potřebujete pomoc, kontaktujte nás).

<b>⊎</b> =			D		Stopped toolcoord0	wobj0 exaxis0	10			12 8
initialize <	📹 🖪 🛎			Goto	X Operation&Status 360	Free Mounting	and Meanil	Ing		
📄 Teaching 🗸	Logic commant ~	C Process1A.lua	example	s1::do-s1 tag header	Move (m)		Robot P	050		(±)
Program Teachi	00 T	1++		PTP.P1,100-execution content end-s1 tag tail	REM			195 .12	79.458	Joints
Graphical Program	While IT_Else	2++ SetWObjCoord(1,464.590,820.900);		goto s1-Jump to s1 tab	× 78*2		J4 :-107.8	164 JB : 4	51.091 JG	: 39.719
Manage Teaching	Goto Wait	3⊷ SetToolCoord(2,-2.600,0.000,644.0	Program				X 1990 N	01 V :4	71.009 Z	TCP : 546 499
4. Status <	III 🕣	4↔ MoveJ( 0.670, -55.310, 134.370, -16					RX:-161.3	196 HEY: 7	2.646 RZ	1909.366-12
88 Auxiliary <	Pause Dolle	5++ SPLCSetDO(1,1)					Tote			FT
🛠 Settings	Var	6++ SPLCSetAO(0,10)		Add	-79.428		Fx:0.000 Tx:0.000	Fy I B Ty I D	000 Fz	: 0.000 : 0.000
		7↔ SPLCSetAO(1,0)	Added Commands		107.000	4	Act_State:			
		B⊷ WaitDI (1,1,0,2)			1.01.801	-	Nort : D	0.0.0		Line-Num
		9⊷ SPLCSetDO(1,0)		Apply	39 719					CiriBox
		10- SPLCSetDO(7, 1)					000 004	DO1 DO5	D02 D06	DD3 D07
		11⊷ WaitMs(20)		Profix	400		CO4	0 005	0006	COT
		12 SPLCSetDO(7, 0)		Sensor			Di0 Di4 Ci0	DIS	D12 D18 C12	DI3 Di7 CI0
		13⊷ SetDO(11,1,0,0)					CCIA	CIE	CIE	Gen

Obrázek 4.7-4-2 Rozhraní příkazu Goto

### 1.3.5.7.4.4 Příkaz Wait

Kliknutím na ikonu "Wait" vstoupíte do rozhraní pro úpravu příkazu Wait.

Tato instrukce je zpožďovací instrukce, která je rozdělena na tři části: "WaitMs", "WaitDI" a "WaitAI".

Příkaz "WaitTime" Jednotkou čekací doby je milisekunda, zadejte počet milisekund, které chcete čekat, klikněte na "Add", "Apply".



Obrázek 4.7-4-42 Rozhraní příkazu WaitTime

"WaitDI", tj. jediné čekání DI, vyberte číslo portu IO, na který se má čekat, stav čekání, maximální dobu čekání a metodu zpracování čekací doby a klikněte na tlačítko "Add" a "Apply".

<i>≣</i> ⊼		<b>G</b>   <b>D</b> (	1							S	top	ped toolcoord0 wobj0	exaxis0	10			12 8
initialize <	± ه 📽			-	• •• •	Wait	1			×	9	peration&Status 360" Free Mov	ming Fo	and Majuri	Ing		
📄 Teaching 🗸	Logic commant~	Process1A.lua	Singl	e di wa	aiting							) In the second		Robot P	ose		(±)
Program Teachi	00 T	1++	Port	ŧ				Ch	1-DI0	۲	-11					70 #59 - 79	atnioL
Graphical Program	While IT_Else	2+++ SetWOb)Coord(1,464,500,820.000	Stat	te				Fa	be .	~		*/902		J4 1-107	864 16:4	51.091 JG	39.719
Manage Teaching	Goto Wat	3⊷ SetToolCoord(2,-2.600,0.000,644	Max	timum	time					ms				× 1991 1	04 Y 54	71 509 2	TCP
4. Status <	II 🕣	.1⊷ MoveJ( 0.670, .65.310, 134.370, -1	Wait	t for tir cessing	meout 9			St	sp error repo	ortir 🛩			-	RX:-151	395 HY: J	2.646 HZ	1-00.909
BB Auxiliary <	Pause Dolle	5+++ SPLCSetDO(1,1)									₽	-138.897	-	Tota			FT
🗙 Settings	Var	0⊷ SPLCSetAO(0,10)							Ad	a	Ð	-79.428		Fx :0.000 Tx :0.000	Ey II Ty 10	000 Fz	10.000 10.000 0.1
		7⊷ SPLCSetAO(1,0)	Multi	i di wai	it.						Ð	75		Act_State	ð.		
		B⊷ WaitDi (1,1,0,2)	Con	dition	al selection			An	d	~	Ð	-107.864	1	Num : D	0.0.0		Line-Num
		9⊷ SPLCSetDO(1,0)		D10	False	~	0	C10	Falsa	¥	P	-91 891					ChiBox
		10⊷ SPLCSetDO(7, 1)	D	DI1	Falso	~	0	CI1	Falso	~	9	39.719		DO0 DO4	DO1	002	DDa
		t1⊷ WaitMs(20)		DI2	False	-	0	CI2	Felse	*				C00 C04	001 005	002	000
		12 SPLCSetDO(7, 0)		DI3	False	~ 1	0	CI3	False	*		Add		Dia	DH	012	DIS DIT
		13⊷ SetDO(11,1,0,0)		DI4	False	-		CI4	Falsa	×	~			C10 C14	CH	CI2 CI6	C() C(7
				DI5	Fatse	4	n.	C15	False	×				-			v

Obrázek 4.7-4-5 Příkazové rozhraní WaitDI

Příkaz "WaitMultiDI", tj. čekání na více DI, nejprve vyberte podmínky zřízení více DI, poté zaškrtněte port DI a stav, na který je třeba čekat, a nakonec nastavte maximální dobu čekání a metodu zpracování čekací doby, klikněte na tlačítko "Add" a "Apply".

ā ≡		<b>() ()</b>						Ad	d .	opped toolcoord0 wobj0	exaxis0 10 🖌 🔬 🔞 🕫 🔗
Initialize <	📹 b 🕹		Mult	i di w	ait					Operation&Status 360" Free Mou	ming Fixed Mounting
📄 Teaching 🗸	Logic commant~	Process1A.lua	Cor	ditio	nai selectic	'n	Ad	id	Ŷ	Move (	Robot Pose
Program Teachi		1++	0	D10	False	2	C10	False	~	RCM	Jointe
Graphical Program	While II_Else	2++ SetWOb)Coord(1,464.500,820.000		DII	False	~	CI1	False	~	**	J1 1-158 897 J2 179 438 J3 175 J4 1-107 864 J6 1-51 891 J6 139 719
Manage Teaching	<u>∧</u> ⊙	3↔ SetToolCoord(2,-2.500,0.000,644,	0	DI2	False	~	CI2	False	~		TCP
∙ -4 Status <	Ш <del>Э</del>	i⊷ MoveJ( 0.670, -55.310, 134.370, -1		DI3	False	~	CI3	False	*		X 1350.301 V 1471.009 Z 1546.498 RX1-151.395 RY172.648 RZ1-06.909
88 Auxiliary <	Pause Dolle	5++ SPLCSetDO(1,1)		DI4	False	~	CI4	Falsa	~	-138.897	FT
🗙 Settings	VIV	0++ SPLCSetAO(0,10)	0	D15	False	~	CI5	False	~	79.428	Fx:0.000 Fy:0.000 Fz:0.000
	1020	7⊷ SPLCSetAO(1,0)		D16	False	~	CI6	False	~	75	Act_State: 1
		B⊷ WaitDi (1,1,0,2)		D17	False	2	C17	Falsa	~	-107.864	Line-Num
		B⇔ SPLCSetDO(1,0)	Max	dimun	n time		5		ms	9 -91 891	Not : 0.00.0
		10+++ SPLCSetDO(7, 1)	Wai	t for t cessi	limeout. ng		S	op error repo	etir 🛩	39.719	CtrlBox Doc DO1 DO2 DO3
		t1⊷ WaltMs(20)						-			004 005 006 007 000 001 002 003 004 005 006 007
		12 SPLCSetDO(7, 0)								Add	
		10⊷ SetDO(11,1,0,0)						Ad	d	-	Ci0 Ci1 Ci2 Ci0 Ci4 Ci5 Ci6 Ci7

Obrázek 4.7-4-6 Příkazové rozhraní WaitMultiDI

"WaitAI", vyberte analogové množství, na které se má čekat, hodnotu, maximální dobu čekání a metodu zpracování čekací doby a klikněte na "Add" a "Apply".

⊴ ≡		<b>G D D D</b>	D	St	opped toolcoord0	wobj0 exaxis0	10			12 8
🔿 initialize <	📹 h 🕁		Wei	×	Operation&Status 3	60" Free Mounting Fis	and Mount	ng		
📄 Teaching 🗸	Logic commant~	Process1A.lua	Port	Ctrl-Al0 🛩	Nove (a)		Robot P	ose		(±)
Program Teachi	00 T	1++	condition	> ~	RGM					Joints
Graphical Program	While IT_Else	2++ SetWOb)Coord(1,464.500,820.000,	numerical value	96	*** */9°2		J4 :-107 B	64 16 1-5	at 1091 JG	: 39.719
Manage Teaching	Goto Wat	3↔ SetToolCoord(2,-2.600,0.000,644.0	Maximum time	ms				with the state	er inne litel	TCP
4- Status 🔇	Ш Э	4↔ MoveJ( 0.670, -55.310, 134.370, -16	Wait for timeout	Stop error reportir~			RX:-151.3	95 RY: 1	2.646 RZ	1 -86.909
88 Auxiliary <	Pause Dolle	5+++ SPLCSetDO(1,1)			-138.896	-	Tota			FT
🗙 Settings	Vor	6++ SPLCSetAO(0,10)		Previous	-79.428		Fx :0.000	Fy II Ty 10	000 Fz	10.000
		7↔ SPLCSetAO(1,0)		Ada	75	4	Act_State:	1		
		B⊷ WaitDi (1,1,0,2)	Added Commands:		-107.864	7	Alexandre Sta	0.00		Line-Num
		9⊷ SPLCSetDO(1,0)			-91.891					-
		10⊷ SPLC SetDO(7, 1)		Арріу	39.719		DOO	DOI	002	DD3
		t1⊷ WaitMs(20)		Profix			000 CO4	001	002	C03 007
		12⊷ SPLCSetDO(7, 0)		Foint name	Add		Dia	DH	012	DIS
		10⊷ SetDO(11,1,0,0)		Sensor	-		C10 C14	CH	C12 C16	CIQ CIQ

Obrázek 4.7-4-7 Příkazové rozhraní WaitAI

# 1.3.5.7.4.5 Příkaz Pause

Kliknutím na ikonu "Pause" vstoupíte do rozhraní pro úpravu příkazu Pause.

Tento pokyn je pokyn pro pozastavení. Vložte tuto instrukci do programu. Když program provede tuto in- strukci, robot bude ve stavu pauzy. Pokud chcete pokračovat v běhu, klikněte na tlačítko "Pause/Resume" v ovládací oblasti.

⊻ ≡	2	<b>G D D D</b>	D		Stopped toolcoord0 v	wobj0 exaxis0	10   1			12 8
@ initialize <	± ه ک	28888		Pasue	× Operation&Status 360" 8	Free Mounting Fix	nd Mauril	Ing		
📄 Teaching 🗸	Logic commant~	Process1A.lua	Pause function	No function			Robot P	ose		æ
Program Teachi	00 🛨	1++		Ad	id and a second s		11 1-110	105-12-17	0.458 - P	Joints
Graphical Program	While IT_Else	2++ SetWOb)Coord(1,464.500,820.000;	Added Commands:		7/9/2		J4 1-107 I	164 JB : -5	1.091	: 99.719
Manage Teaching	Goto Wat	3↔ SetToolCoord(2,-2.500,0.000,644.0					w kiego is	MEN N	N 805 2	TCP
4. Status <	Ш Э	d⊷ MoveJ( 0.670, -55.310, 134.370, -16				-	RX:-151.	195 HY: 12	1.646 RZ	1000.000
88 Auxiliary <	Pause Dolle	5++ SPLCSetDO(1,1)			-138.896	-	Total			FT
💥 Settings	1:3 Vir	6⊷ SPLCSetAO(0,10)		.2 <b>— 0</b>			Fx :0.000	Fy III.	000 Fz	0.000
		7↔ SPLCSetAO(1,0)		J3 😄 — 🧿	o- 🕀 75	4	Act_State	1		
		B⊷ WaitDi (1,1,0,2)		.34 🔵 — O	107 864	k				Line-Num
		B++ SPLCSelDO(1,0)		J5 😄 - <b>O</b> -	- 🔂 [-91.891		Murr : D	0.0.9		
		10 SPLCSetDO(7, 1)		.e 🖨 — o	- 🔂 [39.719		DOO	DOI	002	CtriBex DD3
		t1⊷ WaltMs(20)		Profix			004	005	006	001
		12⊷ SPLCSetDO(7, 0)		Foint same	Add		Dia	Dis	012	Dis
		10 SetDO(11.1.0.0)		Sensor	~		Di4 C10 C14	DIS CH CIS	0/8 C/2 C/6	D17 C10 C17

Obrázek 4.7-4-8 Rozhraní příkazu Pauza

#### 1.3.5.7.4.6 Příkaz Dofile

Kliknutím na ikonu "Dofile" vstoupíte do rozhraní pro úpravu příkazů Dofile.

Příkaz Dofile volá interní program řídicí jednotky. Při použití příkazu Dofile je třeba uložit volaný podprogram, hlavní program není třeba znovu ukládat, pokud se nezměnil. Příkaz Dofile podporuje dvouúrovňové volání a je třeba věnovat pozornost dvěma nastavením parametrů. Jedním je úroveň volání a druhým je identifikační číslo volání. V zásadě platí, že stejné ID číslo se nemůže objevit ve stejném programu.

<i>⊾</i> ≡			D		Stopped toolcoord0	wobj0 exaxis0	0 10			12 8
initialize <	📹 🖻 🕁			Dofile	X Operation&Status 360"	Free Mounting Fi	xed Mauri	ing		
📄 Teaching 🗸	Logic commant~	Process1A.lua	Dofile lua file;	Process1Alua	Move @		Robot P	ose		
Program Teachi	00 7	1++	Program preview	SetWDbiCoord/1 454 500 820 00				195-12-13	9 458 33	Joints
	While IT_Else	2++ SetWOb)Coord(1,464.590,820.900,		0,374.000,0.000,90.000,-90.000) SetToolCoord(2,-2.500,0.000,644	7/5*2		.4 :-107	864 JB : 4	51.091 JG	: 39.719
	Goto Wait	3⊷ SetToolCoord(2,-2.500,0.000,644.0		.000,0.000,0.000,0.000,0,0)	<i>a</i>		X 1390.3	04 V :4	71.809 Z	TCP 1 546 498
	II 🕁	d⊷ MoveJ( 0.670, -55.310, 134.370, -16	Which layer is called	first floor	2		RX(-151)	395 RY: 7	2.046 92	21-00.909
	Pause Dolle	5+++ SPLCSetDO(1,1)	ID No		138.890		Tota			FT
🗙 Settings	Var	6++ SPLCSetAO(0,10)	*ID cannot be repeat	ed in the same program	-79.428		Fx :0.000	Fy 10 Ty 10	000 Fz	10.000 10.000
		7⊷ SPLCSetAO(1,0)		Add	75		Act_State	()		
		B⊷ WaitDI (1,1,0,2)	Added Command	5	[-107/864]	6	Num : D	0.0.0		Line-Num
		B++ SPLCSetDO(1,0)								CtriBox
		10⊷ SPLC SetDO(7, 1)			39.719		DO0 DO4	DO1 DO5	DG2 DOG	DO3 DO7
		t1⊷ WaltMs(20)		Apply			000 CO4	001	C02	000 007
		12 SPLCSetDO(7, 0)		Point same	Add		Dia	Dis	012	DIS
		10 SetDO(11,1,0,0)		Sensor	~		C10 C14	CCH	CI2 CIE	CID CIT

Obrázek 4.7-4-9 Rozhraní příkazu Dofile

## 1.3.5.7.4.7 Příkaz Var

Kliknutím na ikonu "Var" vstoupíte do rozhraní pro úpravu příkazu Var.

Tento příkaz je proměnný systémový příkaz, který je rozdělen na dvě části: Definice proměnné Lua, dotaz na proměnnou a přejmenování proměnné Sys, získání hodnoty a nastavení hodnoty. Definice proměnné Lua může deklarovat proměnnou a přiřadit jí počáteční hodnotu a spolupracovat s příkazy while, if-else a dalšími Příkaz dotaz na proměnnou Lua slouží k dotazování na hodnotu názvu vstupní proměnné v reálném čase a jejímu zobrazení ve stavovém řádku. Počet proměnných Sys je pevně stanoven a můžete je přejmenovávat, získávat hodnoty proměnných a nastavovat jejich hodnoty. Hodnoty uložené v této proměnné se při vypnutí systému nevymažou.



Obrázek 4.7-4-10 Příkazové rozhraní Var

# 1.3.5.7.5 Rozhraní příkazů pohybu



Obrázek 4.7-5 Rozhraní příkazů pohybu

### 1.3.5.7.5.1 Příkaz PTP

Kliknutím na ikonu "PTP" vstoupíte do rozhraní pro úpravu příkazů PTP.

Můžete zvolit bod, kterého má být dosaženo, a nastavením doby plynulého přechodu lze zajistit, aby pohyb M om tohoto bodu do dalšího bodu byl plynulý. Zda nastavit posun, můžete zvolit posun na základě základního souřadnicového systému a na základě souřadnic nástroje a vyskočí nastavení posunu x, y, z, rx, ry, rz, specifická dráha PTP je optimální dráha automaticky naplánovaná řídicí jednotkou pohybu, kliknutím na "Add" a "Apply" tento příkaz uložte.



Obrázek 4.7-5-1 Příkazové rozhraní PTP

# 1.3.5.7.5.2 Příkaz Lin

Kliknutím na ikonu "Lin" vstoupíte do rozhraní pro úpravu příkazů Lin.

Funkce tohoto příkazu je podobná příkazu "PTP", ale cesta bodu dosaženého tímto příkazem je přímka.



# Obrázek 4.7-5-2 Příkazové rozhraní Lin

# 1.3.5.7.5.3 Příkaz Arc

Kliknutím na ikonu "Arc" vstoupíte do rozhraní pro úpravu příkazů Arc.

Příkaz "Oblouk" je obloukový pohyb, který zahrnuje dva body. První bod je přechodový bod uprostřed oblouku a druhý bod je koncový bod. Přechodový bod i koncový bod lze nastavit na posun a můžete zvolit posunutí v základním souřadném systému Posun a posunutí na základě souřadnic nástroje a vyskakovací nastavení posunu x, y, z, rx, ry, rz a koncovému bodu lze nastavit poloměr plynulého přechodu pro dosažení efektu plynulého pohybu.



Obrázek 4.7-5-3.7-6 Příkazové rozhraní oblouku

# 1.3.5.7.5.4 Příkaz Circle

Kliknutím na ikonu "Kruh" vstoupíte do rozhraní pro úpravu příkazu Kruh.

Příkaz "Kruh" je pohyb po celém kruhu, který zahrnuje dva body. Prvním bodem je střední přechodový bod 1 plného kruhu a druhým bodem je střední přechodový bod 2 plného kruhu. Přechodový bod 2 lze nastavit jako posunutý. Účinkuje v přechodovém bodě 1 a přechodovém bodě 2.



Obrázek 4.7-5-4 Rozhraní příkazu Circle

# 1.3.5.7.5.5 Příkaz Spirála

Kliknutím na ikonu "Spirála" vstoupíte do rozhraní pro úpravu příkazů Spirála.

Příkaz "Spirála" je spirálový pohyb, který zahrnuje tři body, které tvoří kruh. Na stránce nastavení třetího bodu je několik parametrů včetně počtu otáček spirály, úhlu korekce polohy, přírůstku poloměru a přírůstku směru osy otáčení. Nastavení, počet otáček spirály je počet pohybových kružnic spirály, úhel korekce polohy je poloha na konci spirály a poloha prvního bodu spirály, přírůstek poloměru je přírůstek poloměru každé kružnice a směr osy otáčení je zvětšen. Částka je přírůstek ve směru osy šroubovice. nastavit Zda se má posunout, posun se projeví na trajektorii celé šroubovice.

ā ≡			I		Stopped toolcoord0 wob	0 exaxis0 10 🖢 🗛 🔞 😿 🕏
initialize <	📹 🖿 🕁		series and the series of the s	aral 🗙	Operation&Status 360" Free	Mounting Fixed Mounting
📄 Teaching 🗸	Motion commar~ Motion command	Process1A.lua	Helix mid point 1	1 ~		Robot Pose
Program Teachi	5.	1++	Tool coordinate system:	htest	RCM	Jointe
	PTP LIN	2⊷ SetWOb)Coord(1,464,500,820.000	Workpiece coordinate system	0	2992	J4 :-107.864 J6 : 51.091 J6 : 59.72
	ARC Orde	3⊶ SetToolCoord(2,-2.500,0.000,644.	x	-144.471		TCP X 1350.301 V 1471.589 Z 1546.499
	88	.4↔ MoveJ( 0.670, .55.310, 134.370, .1	Y	409.616		RX:-151.095 RY: 12.046 RZ:-06.909
	Spinal N-Spinal	5++ SPLCSetDO(1,1)	z	48.701	-138.896	FT Too:
💥 Settings	Some N-Some	0⊷ SPLCSetAO(0,10)	RX	-178.523	79.428	Fx:0.000 Fy:0.000 Fz:0.000 Tx:0.000 Ty:0.000 Tz:0.000
	1 🕄	7⊷ SPLCSetAO(1,0)	RY	2.290	75	Act_State: 1
	Weave TFD	B⊷ WaitDi (1,1,0,2)	RZ	-21.817	-107 864	Line-Num
	Offset ServeC	B++ SPLCSetDO(1,0)			9 -91.891	Chiller
	22	10⊷ SPLC SetDO(7, 1)		Nezz	39.72	D00 D01 D02 D03 D04 D05 D06 D01
		t1⊷ WaitMs(20)				CO0 CO1 CO2 CO3 CO4 CO5 CO6 ED7
	DMP WPTrd	12- SPLCSetDO(7, 0)		Apply	Add	DI0 DI1 DI2 DI3 DI4 DI5 DI6 DI7
		10⊷ SetDO(11,1,0,0)		Sensor	~	C10 C11 C12 C10 C14 C15 C16 C17

Obrázek 4.7-5-5 Rozhraní příkazů spirály

#### 1.3.5.7.5.6 Příkaz N-Spiral

Kliknutím na ikonu "N-Spiral" vstoupíte do rozhraní pro úpravu příkazů N-Spiral.

Příkaz "N-Spiral" je optimalizovaná verze spirálového pohybu. Tento příkaz potřebuje k realizaci spirálového pohybu pouze jeden bod a konfiguraci různých parametrů. Robot bere aktuální polohu jako výchozí bod a uživatel nastavuje rychlost ladění, zda má být posunut, počet otáček spirály, sklon spirály, počáteční poloměr, přírůstek poloměru, přírůstek směru osy otáčení a směr otáčení. Počet otáček spirály je šroubovice. Počet pohybových kružnic, sklon šroubovice je úhel mezi osou Z nástroje a horizontálním směrem, úhel korekce polohy je poloha na konci šroubovice a poloha prvního bodu šroubovice, počáteční poloměr je poloměr první kružnice a přírůstek poloměru To je přírůstek poloměru každé kružnice, přírůstek směru osy otáčení je přírůstek směru osy šroubu a směr oMotávání je po směru a proti směru hodinových ručiček.

ā =			Ň	Spiral 🗙	opped toolcoord0 wobi0 exaxis	0 10 <b>6 A 18 5</b> A
le initialize 🗸	en a ±		Start of helix	1 v	Operation&Status 360" Free Mounting 1	Fixed Mounting
Teaching V	Motion commar~ Motion command	Process1A.lua	Workpiece coordinate	0	I Move Co	Robot Pose
Program Teachi	5.	1++	x	-144.471		Joints ./1 :-198.896 ./2 : .79.458/3 : .75
Graphical Program	PTP LIN	2⊷ SetWOb)Coord(1,464,590,820.900	Y	409.616		.34 :-107.864 .36 :-51.091 .36 : 39.719
Manage Teaching	ARC Circle	3↔ SetToolCoord(2,-2.500,0.000,644.	z	48.701		TCP X :050.302 V :471.509 Z :546.498
4. Status 🔇	88	d⊷ MoveJ( 0.670, -55.310, 134.370, -1	RX	-178.523		RX:+161.395 RY: 12.648 RZ:+00.909
88 Auxiliary 🔇	Spital N-Spital	5+++ SPLCSetDO(1;1)	RY	2.290	-138.896	FT
🗙 Settings	Some N-Some	0⊷ SPLCSetAO(0,10)	RZ	-21.817	70.428	Fx:0.000 Fy:0.000 Fz:0.000 Tx:0.000 Ty:0.000 Tz:0.000
	2 🐨	7⊷ SPLCSetAO(1,0)	Commissioning speed	100 %	75	Act_State: 1
	Weave TFD	B⊷ WaitDI (1,1,0,2)	Offset or not	Tool coordinate offset 🗸	-107 864	Line-Num
	Caller Server	9⊷ SPLCSetDO(1,0)	Number of helices	5	-91 891	Chillor
	1 1	10+++ SPLC SetDO(7, 1)	Helix inclination	30	39.719	000 001 002 003
	Tractory Tractory	t1⊷ WaitMs(20)	Initial redius	50 mm		CD0 CD1 CO2 CO3 CO4 CO5 CO6 CD7
	DMP WPTtsf	12 SPLC SetDO(7, 0)	Radius increment	10 mm	Add	DI0 DI1 DI2 DI3
		10⊷ SetDD(11,1,0,0)	Axis direction increment	10 mm		CI0 CH CR CD CH CR CR CD
			Direction of rotation	Clockwise 🛩		a set and a set of the

Obrázek 4.7-5-6 Příkazové rozhraní N-Spiral

#### 1.3.5.7.5.7 Příkaz Spline

Kliknutím na ikonu "Spline" vstoupíte do rozhraní pro úpravu příkazů Spline.

Příkaz je rozdělen na tři části: začátek skupiny splajnů, segment splajnu a konec skupiny splajnů. Začátek skupiny spline je počáteční značka pohybu spline. Segment spline zahrnuje segmenty SPL, SLIN a SCIRC. Kliknutím na příslušnou ikonu vstoupíte do rozhraní příkazu Add (Přidat), konec spline skupiny je koncovou značkou pohybu spline.



Obrázek 4.7-5-7 Příkazové rozhraní Spline

# 1.3.5.7.5.8 Příkaz N-Spline

Kliknutím na ikonu "N-Spline" vstoupíte do rozhraní pro úpravu příkazů N-Spline.

Tato instrukce je optimalizační instrukcí pro instrukční algoritmus Spline a v budoucnu nahradí stávající instrukci Spline. Tato instrukce je rozdělena na tři části: začátek vícebodové trajektorie, segment vícebodové trajektorie a konec vícebodové trajektorie. Značka začátku, segment vícebodové dráhy slouží k nastavení jednotlivých bodů dráhy, kliknutím na ikonu vstoupíte do rozhraní pro přidávání bodů, konec vícebodové dráhy je značka konce pohybu vícebodové dráhy, zde můžete nastavit režim řízení a rychlost ladění, režim řízení se dělí na Pro daný řídicí bod a daný bod dráhy.



Obrázek 4.7-5-8 Příkazové rozhraní N-Spline

# 1.3.5.7.5.9 Příkaz Weave

Kliknutím na ikonu "Weave" vstoupíte do rozhraní pro úpravu příkazů Weave.

Příkaz "Weave" se skládá ze dvou částí. V první části vyberte číslo tkalcovského stavu s nastavenými parametry, klikněte na tlačítka "Spustit tkaní" a "Zastavit tkaní" a použijte pro přidání souvisejících příkazů do programu.

£ ≣	3	<u>ب</u> ا ک	I	St	opped toolcoord0 wobj0 exaxis	10   10   10   10   10   10
Initialize <	📹 h 🕁	11111111111111111111111111111111111111	Weava	×	Operation&Status 360" Free Mounting Fr	and Mounting
📄 Teaching 🗸	Motion commar~	Process1A.lua	Select No.	0 ~		Robot Pose
Program Teachi	50	3↔	Swing type	Planar sine wave sw	RGM	Joints
Graphical Program	PTP LIN	2++ SetWOb)Coord(1,464.500,820.000	Swinn amplitude	5 00000 mm	7392	J4 :-107.864 J5 :-61.091 J5 : 99.719
Manage Teaching	ARC Orde	3↔ SetToolCoord(2,-2.600,0.000,644.	Swing left dwell time	300.000000 ms		TCP X :350.304 V :471.005 Z :546.499
-↓ Status <	88	i⊷ MoveJ( 0.670, -55.310, 134.370, -1	Swing right dwell time	300.000000 ms		RX:+151.395 RY:12.646 RZ:+66.909
88 Auxiliary <	C C	5+++ SPLCSetDO(1,1)	Configure		-138.897	FT
🛠 Settings	Some N-Some	6⊷ SPLCSetAO(0,10)	re simulation	of swing simul		Fx:0.000 Fy:0.000 Fz:0.000 Tx:0.000 Ty:0.000 Tz:0.000
	Weave TFD	7⊷ SPLCSetAO(1,0)				Act_State: 1
		B⊷ WaltDI (1,1,0,2)	Weave Solution	<b></b>	-10/ 801	Nim : 000.0
	Offset ServeC	9++ SPLCSetDO(1,0)	Start swinging	Stop swinging	39.718	CtriBox
	Trajectory TrajectoryJ	10⊷ SPLCSetDO(7, 1)				000 001 002 003 004 005 006 007 000 001 002 003
	다 다	t1⊷ WaitMs(20)	Added Commands		Add	CO4 CO5 CO5 CO7
	DMP WPTtsf	12 SPLCSetD0(7, 0)		_		Dia Dia Dia Dia Dia Dia Dia Dia Cio Cii Ci2 Cio
		1.0++ SetDO(11,1,0,0)		Apply		CI4 CI8 CI8 CI7

Obrázek 4.7-5-9 Příkazové rozhraní Weave

Kliknutím na tlačítko "Configuration and Test" (Konfigurace a test) nastavte parametry svařování. Po dokončení konfigurace můžete otestovat dráhu svařování spuštěním testu svařování a zastavením testu svařování.

⊴ ≡			D		Stopped toolcoord	10 wobj0 exaxis0	0   10			12 8
🐵 initialize <	📹 🖪 🛎		Weave		× Operation&Status	360" Free Mounting Fi	and Mount	ing		
📄 Teaching 🗸	Motion commar~ Motion command	C Process1A.lua	Select No.	0 ~	More Car		Robot P	ose		-
Program Teachi	5.	1++	Swing type	Planar sine wave 🗸					0.459	Joints
Graphical Program	PTP LIN	2++ SetWOb)Coord(1,464.500,820.000,	Wobble frequency	1.000000 H	tz		J4 (-107 I	64 .16 : -5	1.091	. 39.719
Manage Teaching	ARC Orde	3↔ SetToolCoord(2,-2.600,0.000,644.0	Swing amplitude	5.000000 1	nm 🗍 👘		w keep a	with the state	r4 8215 T	TCP
4. Status <	88	1→ MoveJ( 0.670, -55.310, 134.370, -16	Swing left dwell time	300.000000 11	ns	-	RX:-151	195 HTY: 13	1.648 RZ	1 - 66 909
88 Auxiliary <	Spital N-Spital	5+++ SPLCSetDO(1,1)	Swing right dwell time	300.000000 "	ns	-	Tota			FT
🗙 Settings	Spine N-Spine	0++ SPLCSetAO(0,10)	Return	Configure	-79.428		Fx :0.000	Fy II. Ty II	000 Fz	10.000
	۲ 🐒	7⊷ SPLCSetAO(1,0)			75	6	Act_State	1		
	Weave TFD	B⊷ WaitDi (1,1,0,2)		Apply	-107 864	76	Alien - D	000		Line-Num
	Offset ServeC	9⊷ SPLCSetDO(1,0)		-	-91.891		and a la	ance.		-
	55	10⊷ SPLC SetDO(7, 1)		- <b>o</b> - 🖨 84	39.719		DOG	DOT	DO2	DD3
	Trajectory TrajectoryJ	t1⊷ WaltMs(20)		Profix			C04	001	C02	001
	DMP WPTISE	12⊷ SPLCSetDO(7, 0)		Foint same	Add		DIO	DH	012	DIS
		10⊷ SetDO(11,1,0,0)		Sensor	~		C10 C14	CH	C12 C16	CIQ

Obrázek 4.7-5-10 Konfigurační a testovací příkazové rozhraní Weave

#### 1.3.5.7.5.10 Příkaz TPD

Kliknutím na tlačítko "TPD" vstoupíte do rozhraní pro úpravu

příkazu TPD V tomto příkazu musí mít uživatel nejprve nahranou

stopu.

O nahrávání stop: Před přípravou na nahrávání stopy nejprve uložte počáteční bod stopy. Když je robot v režimu přetahování, zadejte název souboru, vyberte periodu (za předpokladu, že hodnota je x, tj. zaznamenávat bod každých x milisekund, doporučuje se zaznamenávat bod každé 4 milisekundy), bod se začne zaznamenávat a uživatel může přetahovat robota, aby určil Pohyb, po dokončení záznamu kliknutím zastavte záznam a předchozí pohyb

je možné ukládat stopy robota. Pokud pohyb nelze plně zaznamenat, zobrazí se upozornění, že počet bodů záznamu překročil limit a uživatel musí cvičení zaznamenat na několikrát.

Při programování nejprve pomocí instrukce PTP dosáhněte počátečního bodu příslušné trajektorie, poté vyberte trajektorii v instrukci pro reprodukci trajektorie TPD, vyberte, zda je hladká, nastavte rychlost ladění a postupným kliknutím na tlačítka "Add" a "Apply" vložte program. Příkaz pro načtení trajektorie se používá hlavně k předběžnému načtení souboru trajektorie a jeho extrakci do příkazu trajektorie, který se lépe aplikuje na scénu sledování dopravníkového pásu.

**Poznámka:** Podrobné informace o ovládání funkce TPD naleznete v návodu k obsluze modulu výukového programování (TPD).

⊴ ≡	8	<b>()</b>	IJ		Stopped toolcoord0 wobj0 exexis0 10 <b>b</b> 🛦 🔞 🕫						
@ initialize <	<b>61 b</b> ±				Coperation&Status 360" Free Mounting	Fixed Mounting					
📄 Teaching 🗸	Motion commar~ Motion command	Process1A.lua	Track loading			Robot Pose					
Program Teachi	5.	1++	Track name:	轨迹01		Joints					
Graphical Program	PTP LIN	2++ SetWOb)Coord(1,464.500,820.000,		Add		J4 :-107.864 J6 : 51.091 J6 : 59.719					
Manage Teaching	ARC Orde	3↔ SetToolCoord(2,-2.500,0.000,644.0	Track manufactures			TCP					
4. Status 🔇	88	.4→ MoveJ( 0.670, -55.310, 134.370, -16	Track name:	A1		RX:-151.095 RY: 12.646 RZ:-66.909					
88 Auxiliary <	Spital N-Spital	5+++ SPLCSetDO(1,1)	Smooth Track	faise	-138.890	FT					
🗙 Settings	Solite N-Spine	6++ SPLCSetAO(0,10)	Commissioning speed	25	-79.428	Fx:0.000 Fy:0.000 Fz:0.000 Tx:0.000 Ty:0.000 fz:0.000					
	۲ 🕱	7↔ SPLCSetAO(1,0)			75	Act_State: 1					
	Weave TFD	B⊷ WaitDi (1,1,0,2)		Add	-107/864	Line-Num					
	Offset ServeC	9++ SPLCSetDO(1,0)	Added Commands:		-91.891	AND . LALO					
	5 5	10 SPLC SetDO(7, 1)			39.718	DOC DOI DO2 DO3					
	Trajectory Trajectory	t1⊷ WaitMs(20)		Apply		CO0 CO1 CO2 CO3 CO4 CO5 CO6 CO7					
	DWE MELINE	12- SPLC SetDO(7, 0)		Foint same	Add						
		10⊷ SetDO(11,1,0,0)		Sensor	<b>~</b>	CO CH C2 CD C4 C8 C6 C7					

Obrázek 4.7-5-11 Příkazové rozhraní TPD

# 1.3.5.7.5.11 Příkaz Offset

Kliknutím na ikonu "Offset" vstoupíte do rozhraní pro úpravu příkazu Offset.

Tento příkaz je příkazem celkového posunu. Zadejte každý posun, přidejte do programu příkaz pro otevření a zavření a příkaz pohybu mezi začátkem a zavřením bude posunut na základě základních souřadnic (nebo souřadnic obrobku).

ā ≡								topped toolcoor	d0 wobj0 exax	s0   10			12 8
Initialize <	📹 🖿 🛎				Offset		×	Operation&Status	360" Free Mounting	Fixed Mount	ling		
📄 Teaching 🗸	Motion command	Process1A.lua	∆ <b>x</b> [0		20	¢ 0	mm	Move (a.)		Robot P	ose		-
Program Teachl	5.	1++	Δ <b>γ</b> [0	mm	۵ŋ	0	mm	HILL N		11 1 4 1 10	195-19-13	0.458	Joints
Graphical Program	PTP LIN	2++ SetWOb)Coord(1,464.500,820.000,	∆ <b>z</b> [0	mm	Arz	e 0	mm	7/8*2		JH 1-107	B64 J61-8	1.091 JG	: 99.719
Manage Teaching	ARD Circle	3⊷ SetToolCoord(2,-2.600,0.000,644.0	Offset Eng				Offset On	<b>1</b> *		W KRAD	on v sa	71 805 2	TCP
.4. Status ≮	88	i↔ MoveJ( 0.670, -55.310, 134.370, -16		e san an					-	RX:+151	395 HEY: T	2.046 HZ	808.86-12
留 Auxiliary く	Spiral N-Spiral	5+++ SPLCSetDO(1,1)	Added Cor	nmanos:				-138.896		Tota			FT
🗙 Settings	Solito N-Spine	0++ SPLCSetAO(0,10)						-79.428		Fx :0.000	Ey a li Ty z 0	000 Fz	10.000
	٠ 🕱	7⊷ SPLCSetAO(1,0)					Apply	75		Act_State	1		
	Weave TFD	B⊷ WaitDi (1,1,0,2)			of the second se	.14 😑	-0-	-107 864	70	Alian + D	000		Line-Num
	Offset ServeC	9⊷ SPLCSetDO(1,0)				J5 🗲	-0-	91.891					Children .
	<b>T T</b>	10 SPLCSetDO(7, 1)				16 <b>G</b>	-0-6	39.719		000	DOT	002	DDS
	Trajectory TrajectoryJ	t1⊷ WaitMs(20)				Profix				000	001	002	001
	DMP WPTIST	12⊷ SPLCSetDO(7, 0)				Foint sar	ne	Add		Dia	DH	012	C DIS
		10⊷ SetDO(11,1,0,0)				Sensor		~		CIO CIO CIA	CH	C12 C16	CIQ

3.7-5-12 Rozhraní příkazu Offset

#### 1.3.5.7.5.12 Příkaz ServoCart

Kliknutím na ikonu "ServoC" vstoupíte do rozhraní pro úpravu příkazů ServoCart.

Příkaz ServoCart pro servořízení (pohyb v kartézském prostoru), který může řídit pohyb robota pomocí absolutního řízení polohy nebo na základě aktuálního posunu polohy.

<i>ъ</i> ≡			Se	ervoCart	×	opped toolcoord0 wobj0 exaxis	0   10			12 8
@ initialize <	📹 🖪 🛎	±∎∎ ∎ ⊁ €	Sports mode	Absolute position	~	Operation&Status 360" Free Mounting	Fixed Mount	ng		
📄 Teaching 🗸	Motion commar~	🗅 323.lua	x	0	men	Move (ch.)	Robot P	ose		(III)
Treasurement	Motion command		Y	0	mm	RCM				
Program Teachi	PTP LIN	1 MireSearchStart(0,10,10,0,1) 2 Lin(1,100,-1,0,0)	z	0	mm	-	at 1-198.8	66 12 : -7 64 16 : -5	9.438 .19	Joints 175 199 719
Graphical Program	0 6	3 Lin(1,100,-1,0,0)	RX	0	men	2/992				
Manage Teaching	ARD Orde	4 WireSearchEnd(0,10,10,0,10,	RY	0	men		X 1060.0	00 V :41	71.009 Z	TCP : 545 498
4. Status <			RZ	0	mm		RX:-151.2	196 HY: 1	2.646 RZ	1996.309
88 Auxiliary <	Spiral N-Spiral		Scale factor x	0	1	-138.890	Tota			FT
🗙 Settings	Some N-Spine		Scale factor y	0		70 428	Fx :0.000 Tx :0.000	€y : 0 Ty ± 0	000 Fz 000 Tz	0.000 10.000
	X 🖣		Scale factor z	0			Act_State:	3) 		
	Neave THD		Scale factor rx	0		-107 864	Nut : 0	0.00		Line-Num
	Offset ServeC		Scale coefficient ry	0	-	9 -91.891				-
	<b>T T</b>		Proportional coefficient	0		39.718	DOG	DOT	002	DOS
	Tractory TractoryJ		rz				000	000	000	001
	CD CD		Acceleration	180	95		CO4	0005	0006	EOT.
	DMP WPTIst		Speed	100	46	2031	D10 D14	DIS	D12 D19	DIS DI7
			Instruction cycle	0	9		C10 C14	CIE	C/6	C07

Obrázek 4.7-5-13 Příkazové rozhraní ServoCart

Příklad programu pro řízení absolutní polohy

```
1 PTP(p1,100,0,0)
2
   x,y,z,rx,ry,rz = GetActualTCPPose()
3 * while 1 do
4 +
       if type(x) == "number" then
5
           x = x+1
6
           WaitMs(8)
7
           ServoCart(0,x,y,z,rx,ry,rz,0,0,0,0,0,0,180,100,0.008,0,0)
8
       end
9
   end
```

V tomto příkladu jsou x, y, z, rx, ry, rz (kartézská poloha) aktuální poloha robota. Kromě toho může uživatel řídit pohyb robota načtením souboru dat trajektorie a odesláním dat trajektorie prostřednictvím komunikace přes socket.

Příklad řídicího programu založeného na aktuálním posunu polohy (posun základní souřadnice):



# 1.3.5.7.5.13 Příkaz Trajctory

Kliknutím na ikonu "Trajctory" vstoupíte do rozhraní pro úpravu příkazů Trajctory.

<i>≣</i> ≣	2	$\odot$ $\odot$	0	Stopped toolcoord0 wobj0 exaxis	s0 17 <b>6 A 8 17 8</b>
initialize <	± ه ک		E Trajctory	Coperation&Status 360" Free Mounting 1	Fixed Mounting
📄 Teaching 🗸	Motion command	🗅 081415.lua	Import Trajectory File	Move Ca.	Robot Pose
Program Teachi	5	3++	选择文件未选择文件		Joints 10:00:01:01:00:00:01:01:00:00:00:00:00:0
	PTP LIN	2+++ PTP(arc1,100,-1,0)	Import	-/s*2	J4 1-137 664 J6 1-89 745 J6 145 979
Manage Teaching	ARD Orde		Preload Trajectory		TCP X : 276.039 V : 463.072 Z : 166.828 HX: 179.574 HV: 0.037 HZ: 00.574
-4- Status <	Spinal N-Spinal		Select trajectory file	44.402	FT
* Settings	2 D		Add	70.789	Tour Fx:0.000 Ey:0.000 F2:0.000
	1		Trajectory Motion	117.597	Tx (0.000 Ty 10.000 Te 10.000 Act_State: 1
	Weave TFD		Commissioning speed	-137 684	Line-Num
	Citizet ServeC		A02	89 745	ChiBox
	Trajectory Trajectory		Print Trajectory Point Id	142MI3	000 001 002 003 004 005 006 007
	다		Add	Add	
	DMP WPTist		Added Commands		Di4 Di5 Di6 Di7 Ci0 Ci1 Ci2 Ci0
					CIA CIE CIE CIF

Obrázek 4.7-5-14 Příkazové rozhraní Trajctory

#### 1.3.5.7.5.14 Příkaz TrajctoryJ

Kliknutím na ikonu "TrajctoryJ" vstoupíte do rozhraní pro úpravu příkazů TrajctoryJ.

1. Funkce importu souboru trajektorie: výběr lokálního počítačového souboru pro import do řídicího systému robota.

2. Přednahrávání stopy: vyberte importovaný soubor stopy a načtěte jej příkazem

3. Pohyb po trajektorii: Odeslání pohybu robota prostřednictvím kombinace příkazu přednahraného souboru trajektorie a zvolené rychlosti ladění.

4. Tisk čísla bodu dráhy: tisk čísla bodu dráhy během jízdy robota po dráze, aby bylo možné zkontrolovat průběh aktuálního pohybu.



Obrázek 4.7-5-15 Příkazové rozhraní TrajctoryJ

# 1.3.5.7.5.15 Příkaz DMP

Kliknutím na ikonu "DMP" vstoupíte do rozhraní pro úpravu příkazů DMP.

DMP je metoda učení napodobováním trajektorií, která vyžaduje předchozí plánování oMeference trajektorií. V rozhraní pro úpravu příkazů , vyberte učící bod jako nový výchozí bod, klikněte na tlačítko "Add" a "Apply" pro uložení příkazu. Specifická dráha DMP je nová trajektorie, která napodobuje referenční trajektorii s novým výchozím bodem.

⊴ ≡				s	Stopped toolcoord0 wobj0 exaxis	50   17 <b>  6   A</b>   🕲   🛫   O
@ initialize <	📹 🗅 🛎		Dr	MP X	Operation&Status 360" Free Mounting 1	Fixed Mounting
📄 Teaching 🗸	Motion commar~	🗅 081415.lua	Point Name:	arct 🗸		Robot Pose
Program Teachi	5.	3++	Tool coordinate system:	teoicoord0	RCM	Joints
Graphical Program	PTP LIN	2++ PTP(arc1,100,-1,0)	Workpiece coordinate system	0	2303	J4 :-137.004 J6 : 46.745 J6 : 45.973
Manage Teaching	ARC Circle		JI	45.248		TCP X 1-276 037 V 1-463.074 Z 1166.829
4. Status 🔇			JZ	-67.835		RX:-179.574 RY:0.007 RZ:00.524
BB Auxiliary <	Spital N-Spital		13	117.534	44.402	FT
🗙 Settings	Some N-Spine		J4	-139.791	-70.789	Fx:0.000 Fy:0.000 Fz:0.000 Tx:0.000 Ty:0.000 Tz:0.000
	1 1		J5	-92.493		Act_State: 1
	Neave 140		16	37.396	-137 684	Num : 0.0.0.0
	Offset ServeC		Commissioning speed	[100 ] %	89.745	ChiBox
	Traectory Traectory			Add	45.973	000 001 002 003 004 006 006 007
	61 62		Added Commands			CO4 CO5 CO6 CO7
	DMP WPTrsf				Add	DI0 DI1 DI2 DI3 DI4 DI5 DI0 DI7
				Apply		Ci4 Ci5 Ci6 Ci7

Obrázek 4.7-5-16 Příkazové rozhraní DMP

# 1.3.5.7.5.16 Příkaz WPTrsf

Kliknutím na ikonu "WPTrsf" vstoupíte do rozhraní pro úpravu příkazů WPTrsf.

Vyberte souřadnicový systém obrobku, který má být automaticky převeden, a kliknutím na tlačítko "Add" a "Apply" uložte in- strukci. Tato instrukce realizuje automatický převod bodů v souřadnicovém systému obrobku při provádění interních instrukcí PTP a LIN. Pomocí oblasti příkladů si zobrazte a vyzvěte správnou kombinaci instrukcí. Po přidání konkrétních instrukcí můžete kombinaci upravit podle aktuální scény.

⊴ ≡		<b>G</b>   <b>D</b> (	II		Stopped toolcoo	ord0 wobj0 exaxis	0   17   1			12 8
@ initialize <	ن ه <b>ه</b>			WPTref	× Operation& Status	a 360" Free Mounting F	ixed Mauri	Ing		
📄 Teaching 🗸	Motion command	🗅 081415.lua	Configure		Nove Car		Robot P	080		(±)
Program Teachi	5	3↔	Select workpiece coordinate system					2 3237	0.749	Joints
Graphical Program	PTF LIN	2↔ PTP(arc1,100,-1,0)		Ada	2/9*2		34 :-137 (	164 .15 : -0	6.745 Jb	: 45.979
Manage Teaching	ARC Orde		- and the second				X :-276.0	137 V : 4	83.070 Z	TCP : 166.927
4 Status <	Spiral N-Spiral		example	WorkPieceTrsfStart(1) # PTP(A1,100,0,0) PTP(A2,100,0,0)	44 402	100				FT
X Settings	22			PTP(A3,100,0,0) WorkPieceTrstEnd()	70 789		Tota Fix : 0.000	€y∤0.	000 Fz	: a oan
	1 Same			-Put the movement points y to be converted (for	117.597		Act_State	Ty ±0.	000 Te	1.0.000
	Weave TFD				-137 684		-			Line-Num
			Added Commands:		-89 745		Num : 0	0.0.0		
	Offset ServeC				145 973					CtriBex.
	4 4						DO0 DO4	DO1 DO5	DO2 DO6	DO3 DO7
	Trajectory Trajectory			Apply			000	001	002	000
				Fornt name	Add		Dia	Dec.	000	Des
	DMP WPTISE						Di4	DIS	DH	0.07
				Sensar toolcoord	d8 🗸		C C10	CCH	CCI2	CIO
							CIA	CIE	CIE	CIT

Obrázek 4.7-5-17 Příkazové rozhraní WPTrsf

# 1.3.5.7.6 Rozhraní řídicích příkazů

ā ≡		Stopped toolcoord0 wobj0 exaxis	0 10 6 🔬 🖲	
🐵 initialize 🧹 📹 🛅 土	⊈∎∎⊨∌⊁≣≞≙⊽∣≓	Coperation&Status 360" Free Mounting F	lived Mounting	
Teaching V Control comma V	00	Joint Base Tool Wobi Move	Robot Pose	Æ
Program TeachI D A iro Alo Graphical Program V V		Speed 100 % Acceleration 180 %%/2	J1 :-138.895 J2 :-79.438 J4 :-107.864 J5 :-91.891	Joints .13 : 75 .16 : 99.72
Manage Teaching Iria Iria Viri40 Aux40		Threshold 30 * Single Multi	X :050.301 Y :471.509 RX:-161.395 RY112.546	TCP Z : 546.498 RZ: -86.908
HoveDo TeoList		JI - 0	Tool Fx:0.000 Fy:0.000	FT Fz:0.000
Acc			Tx :0.000 Ty :0.000 Act_State: 1	Tz:0000
		J5 🗢 -o 🕂 (91.891	Num : 0.0.0.0	
		J5 😑 —o- 🛟 39.72	006 001 00 004 005 00	CtriBox 2 D03 6 D07
		Profix Point name Add	CO4 CO5 CO	2 000 6 007
		Sensor 👻	Cité Cité Cité	

Obrázek 4.7-6 Rozhraní řídicích příkazů

# 1.3.5.7.6.1 Příkaz IO

Kliknutím na ikonu "IO" vstoupíte do rozhraní pro úpravu příkazů IO.

Příkaz "IO" se dělí na dvě části: nastavení IO (SetDO/SPLCSetDO) a získání IO (GetDI/SPLCGetDI).

"SetDO/SPLCSetDO" Tímto příkazem lze nastavit zadaný stav DO výstupu, včetně 16 digitálních výstupů řídicí jednotky a 2 digitálních výstupů nástroje. Volba stavu "False" je zavřený, "True" je otevřený a zda blokovat volí možnost "blokováno". " Označuje, že stav DO je nastaven po zastavení pohybu, a výběr možnosti "neblokuje" znamená, že stav DO je nastaven během posledního pohybu. Výběr možnosti "přerušit" pro plynulou trajektorii znamená nastavení stavu DO po skončení poloměru plynulého přechodu a výběr možnosti "vážný" znamená nastavení stavu DO během pohybu poloměru plynulého přechodu. Když je tato instrukce přidána do pomocného vlákna, zda je třeba, aby aplikační vlákno vybralo ano, a ostatní místa používají tuto instrukci k výběru ne. Klikněte na tlačítko "Přidat", "Použít".

5 ≣		I I I I I I	D		Stopped toolcoord	0 wobj0 exaxis0	10			12 8
@ initialize <	1 B ±		Sett0		X Operation&Status	160" Free Mounting Fi	and Mouril	Ing		
🔲 Teaching 🗸 🖸	antrol comma ~	Process1A.lua	Port	Ctrl-DO0	likova 🍙		Robot P	ose		-
Program Teachi	D A	1↔	State	False				05:09:03	70,459 - 79	Joints
Graphical Program	IGIA CNI	2++ SetWOb)Coord(1,464.500,820.000)	Whether it is blocked	block	7/9*2		.4 :-107	164 JG : 4	51.091 JG	. 39.719
Manage Teaching	ITO ITO	3↔ SetToolCoord(2,-2.600,0.000,644.0	Smooth Track	Break						TCP
ے۔ Status 🕻	() 🔧	4↔ MoveJ( 0.670, -55.310, 134.370, -16	Apply thread or not	no			RX:-151.	00 V 14 395 HTV: 7	2.646 82	1: -86.909
🔠 Auxiliary <	oveDO TooLHt	5++ SPLCSetDO(1,1)	"The non blocking (spicseble) instact with the motion instruction, and it has	tion needs to be used togethe no effect when executed alor	-138.896		-			FT
🛠 Settings	Mote Collision	6++ SPLCSetAO(0,10)		Next	-79.428		Fx :0.000	Ey all	1000 F2	10.000
	d)	7↔ SPLCSetAO(1,0)		Add	75	2	Act_State	1		
	ACC	B⊷ WaitDI (1,1,0,2)	Added Commands:		-107.864	10	-			Line-Num
		0+++ SPLCSetDO(1,0)			91.891		MITT : D	0.0.0		
		10++ SPLC SetDO(7, 1)		Apply	39.719		DOG	DOT	DG2	CtriBex DD3
		t1⊷ WaitMs(20)		Profix			004	005	006	001 003
		12 SPLCSetDO(7, 0)		Foint same	Add		Dia	DIS	012	Dis
		10⊷ SetDO(11,1,0,0)		Sensor	~		C10 C14	Cit Cit	CI2 CI2 CI6	D/7 C/0 C/7

Obrázek 4.7-6-1 Rozhraní příkazu SetDO

V příkazu "GetDI/SPLCGetDI" vyberte hodnotu čísla portu, kterou chcete získat, zda chcete blokovat nebo ne, vyberte možnost "blokovat" pro získání stavu DI po zastavení pohybu a vyberte možnost "neblokující" pro získání stavu DI během posledního pohybu. Když je tato instrukce přidána v pomocném vlákně, zda aplikační vlákno musí zvolit ano, a ostatní místa používají tuto instrukci pro výběr ne. Po výběru klikněte na tlačítka "Přidat" a "Použít".

<u>v</u> ≡		<b>G (</b>	I)		Stopped toolcoord0	wobj0 exaxis0 1			12 8
@ initialize <	± ه ک		Sel	oi#	X Operation&Status 360	Free Mounting Fixed h	eunting		
📄 Teaching 🗸	Control comma ~	Process1A.lua	Port	Ctrl-Di0	Move Car	Ro	oot Pose		æ
Program Teachi	D A	1++	Whether it is blocked	block			110.005 10.1	70 419 70	Joints
Graphical Program		2++ SetWOb)Coord(1,464.500,820.000;	Apply thread or not	no			-107 864	51.091 J6	: 99.719
Manage Teaching	VILID Aux-ID	3↔ SetToolCoord(2,-2.500,0.000,644.0	This instruction is generally used in igetor (0:==1) if (spicgetor (0:0.000)	or conditional judgment, such as (0)===1)	(7		195 304 V -	171 805 2	TCP
4. Status 🤇	d 🔧	i⊷ MoveJ( 0.670, -55.310, 134.370, -16		Previous		RX	-151 395 HY:	72.646 92	809.86-12
🔠 Auxiliary <	MoveDO TooLBt	5+++ SPLCSetDO(1,1)		Add	-138.896	Tot			FT
🛠 Settings	Mode Collision	6⊷ SPLCSetAO(0,10)	Added Commands:		-79.428	Fx	0.000 Fy:	1.000 Fz	10.000
		7↔ SPLCSetAO(1,0)			75	Act	State: 1		
	ACC	B⊷ WaitDi (1,1,0,2)		Annie	-107 864	10	: 0000		Line-Num
		⊕⊷ SPLCSetDO(1,0)			-91.891				ChiBox
		10++ SPLC SetDO(7, 1)		.6 🔵 —0-	39.719			D02	DOS
		t1⊷ WaltMs(20)		Profix		c c	00 001 04 005	002	C01
		12⊷ SPLCSetDO(7, 0)		Foint same	Add	95	0 DH	0012	C Dia
		13 SetDO(11,1,0,0)		Sensor	~	C C		C/E	CD CD

Obrázek 4.7-6-2 Příkazové rozhraní GetDI

#### 1.3.5.7.6.2 Příkaz Al

Kliknutím na ikonu "AI" vstoupíte do rozhraní pro úpravu příkazů AI.

Tato instrukce je rozdělena na dvě funkce: nastavení analogového výstupu (SetAO/SPLCSetAO) a získání analogového vstupu (GetAI/SPLCGetAI).

"SetAO/SPLCSetAO" vyberte analogový výstup, který je třeba nastavit, zadejte hodnotu, kterou je třeba nastavit, rozsah je 0-10, zda blokovat nebo ne vyberte "blokovat" znamená nastavit stav AO po zastavení pohybu, vyberte "neblokovat" znamená nastavit stav AO po posledním pohybu Nastavte stav AO v procesu. Když je tato instrukce přidána do pomocného vlákna, zda aplikační vlákno musí zvolit ano, a ostatní místa používají tuto instrukci pro výběr ne. Klikněte na tlačítko "Přidat", "Použít".

<i>ī</i> a ≡			D		Stopped toolcoord	0 wobj0 exaxis0	10			12 0
@ initialize <	<b>e a</b> b	▲ 📾 😫 🖉 🖉 🛎	SelA	ю	Operation&Status	160" Free Mounting Fi	red Mauri	ing		
📄 Teaching 🗸	Control comme ~	Process1A.lua	Port	Ctrl-A00	Nove Car		Robot P	080		æ
Program Teachi		1↔	numerical value	[				IGE: 12 1 3	70.458	Joints
Graphical Program		2 SetWOb)Coord(1,464.500,820.000,	Whether it is blocked	block			.4 :-107.1	64 .15 : 4	51.091 JG	1 39.719
Manage Teaching	VIFJD Aux-IO	3↔ SetToolCoord(2,-2.500,0.000,644.0	Apply thread or not	no	-		X 1090.0	0 V :4	71.509 Z	TCP
	c) 🤸	1↔ MoveJ( 0.670, -55.310, 134.370, -16		Nex		-	RX:-151.	195 HY: 1	2.646 HZ	1996.864
88 Auxiliary <	NoveDO Took.Bt	5+++ SPLCSetDO(1,1)		Arte	-138.896		Tota			FT
🛠 Settings	Mote Collision	0++ SPLCSetAO(0,10)	Added Commands		79.428		Fx :0.000 Tx :0.000	Fy:0 Ty:0	.000 Fz .000 Tz	10.000 10.000
	c)	7↔ SPLCSetAO(1,0)			75	8	Act_State	1		
	ACC	B⊷ WaitDl (1,1,0,2)			-107 864	6	Num : D	0.0.0		Line-Num
		0+++ SPLCSetDO(1,0)		App	9 -91.891					CtriBox
		10++ SPLCSetDO(7, 1)		.6 🔵 — 0			1000 1004	DO1 DO5	DG2 DG6	DO3 DO7
		t1⊷ WaitMs(20)		Profix			000 CO4	001 005	002	003 007
		12 SPLCSetDO(7, 0)		Foint same	Add		Di0 Di4	DIS	012	DIS DI7
		10⊷ SetDO(11,1,0,0)		Sensor	~		C10 C14	CH	CI2 CIE	CO

Obrázek 4.7-6-3 Rozhraní příkazu SetAO

"GetAI/SPLCGetAI" vybírá analogový vstup, který je třeba získat, zda blokovat nebo ne vybírá "blokováno".

získat stav UI po zastavení pohybu a vybere možnost "neblokovaný" pro získání stavu UI během posledního pohybu. Když je tato instrukce přidána v pomocném vlákně, zda aplikační vlákno musí vybrat ano, a ostatní místa používají tuto instrukci k výběru ne. Klikněte na tlačítko "Přidat", "Použít".

⊻ ≡			I)		Stopped toolcoord	0 wobj0 exaxis0	10			12 8
initialize <	<b>ස්</b> ම ස්	2 8 8 8 × 8	Se	AO	Operation&Status	360" Free Mounting Fixe	nd Maurill	na		
📄 Teaching 🗸	Control comma ~	Process1A.lua	Port	Ctrl-Al0	- How Co.		Robot Po			-
Program Teachi.		1↔	Whether it is blocked	błock	~				0.000 0	Joints
Graphical Program	GIA OID	2⊷ SetWObjCoord(1,464.500,820.000;	Apply thread or not	no	*/Sol		J4 :-107 B	64 .15 : -5	1.091 JG	: 99.719
Manage Teaching	Vir-JD Aux-ID	3↔ SetToolCoord(2,-2.600,0.000,644.0	"This instruction is generally used	for conditional (udgment)	such as f		w		71 805 2	TCP
4. Status 🔇	C) 🤸	d↔ MoveJ( 0.670, -55.310, 134.370, -16	(geta) (0)>5); if (spicgelai (0,0.510	100)==1) Pro	WIGHT	-	RX:-161.3	95 HLY: 17	2.046 HZ	808.86-12
BB Auxiliary <	MoveDO TooList	5+++ SPLCSetDO(1,1)			-138.896	-	Tota			FT
🛠 Settings	Mote Collision	0⊷ SPLCSetAO(0,10)	Added Commanda		79.428		Fx :0.000 Tx :0.000	Fy II Ty I 0	000 Fz	10.000
	[]	7↔ SPLCSetAO(1,0)	Photo Commands.		75	6	Act_State:	1		
	ACC	B⊷ WaitDI (1,1,0,2)			-107 864		Murr + D/	100		Line-Num
		0+++ SPLCSetDO(1,0)			-91.891					Chiller
		10 SPLC SetDO(7, 1)		.5 🔵 —	0- 🔂 [39.719]		DOG	DOT	D02	DOS
		t1⊷ WaltMs(20)		Profix			000 004	C01	C02	C01 E07
		12 SPLCSetDO(7, 0)		Point same	Add		Dia	Dis	012	C Dia
		10⊷ SetDO(11,1,0,0)		Sensor	~		C10 C14	CH	C12 C16	CIP

Obrázek 4.7-6-4 Rozhraní příkazu GetAI

# 1.3.5.7.6.3 Příkaz Vir-IO

Kliknutím na ikonu "Vir-IO" vstoupíte do rozhraní pro úpravu příkazů Vir-IO.

Tento příkaz je virtuální řídicí příkaz IO, který může realizovat nastavení simulovaného stavu externího DI a AI a získat simulovaný stav DI a AI.

<u>v</u> ≡		<b>G</b>   <b>D</b> (	Virtual-HO 🗙	opped toolcoord0 wobj0 exaxis	0 10 🖌 🛦 🕲 🐋 🛞				
@ initialize <	<b>ම ස</b> ය		Set analog external di	Operation&Status 360" Free Mounting F	ixed Mounting				
📄 Teaching 🗸	Control comma ~	🗅 323.iua	Port Vir-Ctil-Di0 V	How (A)	Robot Pose				
Program TeachL Graphical Program Manage Teaching ↓ Status < BB Auxiliary < ★ Settines	Hove Do	<pre>1 WiretearchStart(0,19,10,0,1) 2 Lin(1,100,-1,0,0) 3 Lin(1,100,-1,0,0) 4 WiretearchEnd(0,10,10,0,10,0)</pre>	Set up analog external ai Port Vir-Cel-Al0 v numerical value virna Adz	1138.000 779428	Joints 11 : - 131.866 . 27 : 76 .03 . 27 : 75 34 : - 407.864 . 45 : 61.097 . 46 : 27 : 9 X : 300.307 V : 471.509 . 2 : 64.646 50: - 161.396 .471 : 72.648 . 422 : 60.303 FT Tor Tor Tor Tor Tor Tor Tor				
	Ę.		Get analog external di		Act_State: 1				
	ACC		This instruction is generally used for conditional judgment, such as if	9 -107.864 <b>2</b>	Nim : 0.0.0				
			(genomous) op=1), it genomousos (op=1)	39.718	CtriBex 000 001 002 003 004 005 006 007 000 001 002 003				
			Get simulated external ai	Add	CO4 CO5 CO6 CO7				
			Port Vir-CithAl0 V	-	Di4 Di5 Di6 Di7 Ci0 Ci1 Ci2 CD				
			"This anshuction is generally used for conditional judgment, such as if (getvirtuation(0)>5). If (getvirtuationiai (0)>1)		C14 C15 C16 C17				

Obrázek 4.7-6-5 Příkazové rozhraní Vir-IO

### 1.3.5.7.6.4 Příkaz Aux-IO

Kliknutím na ikonu "Aux-IO" vstoupíte do rozhraní pro úpravu příkazů Aux-IO.

Aux-IO je instrukční funkce pro komunikaci robota s PLC za účelem ovládání externího rozšiřujícího IO. Je nutné, aby robot navázal komunikaci UDP s PLC. Na základě původního 16kanálového vstupu a výstupu lze rozšířit 128kanálový vstup a výstup. Použití tohoto příkazu je stejné jako výše uvedené. Použití IO je podobné. Při používání této funkce existují určité technické potíže, obraťte se na nás předem o konzultaci.



Obrázek 4.7-6-6 Příkazové rozhraní Aux-IO

#### 1.3.5.7.6.5 Příkaz MoveDO

Kliknutím na ikonu "MoveDO" vstoupíte do rozhraní pro úpravu příkazu MoveDO.

Tento příkaz realizuje funkci kontinuálního výstupu signálu DO podle nastaveného intervalu během lineárního pohybu.

⊴ ≡			D	Stopped toolcoord0 wo	obj0 exaxis0 10			12 8	
@ initialize <	📹 🗅 🕁			MaveDO	× Operation&Status 360" Fro	e Mounting Fixed Mou	ottog		
📄 Teaching 🗸	Control comma ~	🗅 323.lua	Port	Ctrl-D00	J Move @	Robol	Pose		(#)
Program Teachi	D A	1↔ WireSearchStart(0,10,10,0,10,10,0)	Set the interval	[10]m	m	at 1-15	8 895 .12 .1	79.438 .19	Joints
Graphical Program		2⊷ Lin(1,100,-1,0,0)	Output pulse duty cycle	e [50 ]%	2992	JA 5-10	7.864 .16 : 4	51.091 JG	1 39 7 1 9
Manage Teaching	Vir-JD Aux-ID	3↔ Lin(1,100,-1,0,0)		Add		X :39	. 300 V : 4	171.009 Z	TCP 1 546 499
4. Status 🤇	(†) 🔧	4↔ WireSearchEnd(0,10,10,0,10,10,0)	example	MexeD/OChat/1 10 50		RX:-18	1.395 HY: 1	2.646 RZ	1: -86.909
88 Auxiliary <	MoveDO TooList		example	Lin(A1,100,0,0) MoveDOSton()	-138.890	Tota			FT
🛠 Settings	Mode Collision			invisuosity)	-79.428	Fit 10.0 TX 10.0	00 Ey 10 00 Ty 10	.000 Fz 1.000 Tz	10.000
	ACC .					ACI_SI	1000 I		
					[-107.004]	Nut :	0.0.0		Line-Num
			Added Commands:		30,730				CiriBox
						DO0 DO4	DO1 DO5	DG2 DG6	DO3 DO7
				Apply		CON	005	002	CEDT
					Add	Di0 Di4	Dis	DI2 DHI	DIS DI7
				Sensor	~	C10 C14	Cit.	C16	CIT

Obrázek 4.7-6-7 Rozhraní MoveDO

### 1.3.5.7.6.6 Příkaz ToolList

Kliknutím na ikonu "ToolList" vstoupíte do rozhraní pro úpravu příkazů ToolList.

Vyberte název souřadnicového systému nástroje a kliknutím na tlačítko "Použít" přidejte tento příkaz do programu. Když program spustí tento příkaz, nastaví se souřadnicový systém nástroje robota.

⊴ ≡			Stopped toolcoord0 wobj0 exaxis0 10 🖢 🛦 🔞 🛫 S								
initialize <	📹 🗅 🛎		ToolLet		× Operation&Status 36	O' Free Mounting Fix	and Mount	ing			
📄 Teaching 🗸	Control comma ~	Process1A.lua	Tool coordinate system	toolcoord0 .	More Car		Robot P	ose			
Program TeachI	D A	1++	name	Add				195 - 12 7	9.458 .19	Joints 175	
Graphical Program		2++ SetWOb)Coord(1,464.500,820.000;			*/s*2		J4 :-107 I	164 JB 1-5	dL 1091 J6	: 39.719	
Manage Teaching	Vir-ID Aux-ID	3↔ SetToolCoord(2,-2,600,0.000,644.0	Workpiece coordinate system name	webjcoord0	3		X 1360.3	01 V :4	71.009 Z	TCP 1 546 498	
4. Status 🤇	4° C	d↔ MoveJ( 0.670, -55.310, 134.370, -16		Add			RX:+151.)	195 FEY: 1	2.646 RZ	1-86.909	
88 Auxiliary K	MoveDO TooLBt	5+++ SPLCSetDO(1,1)		-	-138.896		Tote			FT	
🛠 Settings	Mote Collision	0⊷ SPLCSetAO(0,10)	Added Commands:		-79.428		Fx : 0.000 Tx : 0.000	Fy 10 Ty 10	000 Fz	: 0.000 : 0.000	
	ţ.	7↔ SPLCSetAO(1,0)			75	<u> </u>	Act_State				
	ACC	8⊷ WaitDi (1,1,0,2)		Apply	-107 864	-	Not : D	0.0.0		Line-Num	
		9⊷+ SPLCSetDO(1,0)			-91.891					ChiBox	
		10++ SPLCSetDO(7, 1)		JE 🔵 — O-	39.719		D00 D04	DO1 DO5	DO5	DO3 DOT	
		t1⊷ WaitMs(20)		Profix			000 CO4	C01 C05	002	001	
		12⊷ SPLCSetDO(7, 0)		Point same	Add		Di0 Di4	DIS	012	DIS DI7	
		10⊷ SetDO(11,1,0,0)		Sensor	~		Ci0 Ci4	CCH	C/E	C0 C0	

Obrázek 4.7-6-8 Rozhraní příkazu ToolList

### 1.3.5.7.6.7 Příkaz Mode

Kliknutím na ikonu "Mode" vstoupíte do rozhraní pro úpravu příkazů Mode.

Tento příkaz může přepnout robota do ručního režimu a obvykle se přidává na konec programu, aby uživatel mohl robota automaticky přepnout do ručního režimu a po spuštění programu robota přetáhnout.

ā ≡			Stopped toolcoord0 wobj0 exaxis0 10 🖢 🛦 🔞 ⊄ ⊗								
@ initialize <	📹 ն 🛎			Mode	X Operation&Status 360	* Free Mounting Fix	ed Maurill	ng			
📄 Teaching 🗸	Control comma ~	Process1A.lua	Robot mode	Manual mode	V Move Car		Robot Po	58e		-	
Program Teachi	D A	3++		Add	HEM		ut 1-198.0	95 .12 1 .7	9.438 3.3	Joints	
Graphical Program		2++ SetWOb)Coord(1,464.500,820.000,	Added Commands:		"/S <sup>6</sup> 2		J4 :-107.8	64 .16 : -5	1.091 JG	: 99.749	
Manage Teaching	Vir-JD Aux-ID	3⊷ SetToolCoord(2,-2.500,0.000,644.0			<b>.</b>		X 1360.30	a v :41	1.509 Z	TCP	
4. Status <	(j) 🔧	4→ MoveJ( 0.670, -55.310, 134.370, -16		Apply			RX:-161.3	95 RY: 12	1648 RZ	809.86	
88 Auxiliary <	MoveDO TooLH	5++ SPLCSetDO(1,1)			-138.896		Total			FT	
💥 Settings	Note Collision	0⊷ SPLCSetAO(0,10)		2 🔵 — 🗕	79.428		Fx:0.000 Tx:0.000	Fy : 0. Ty : 0/	000 Fz	000 000	
	Ę.	7⊷ SPLCSetAO(1,0)		12 👄 — O	0 75		Act_State:	1			
	ACC	B⊷ WaitDI (1,1,0,2)		.34 🔵 — O	107 864	<u>E</u>	Murr : 0.0	20.0		Line-Num	
		B++ SPLCSetDO(1,0)		J5 😄 – O						CtriBex	
		10⊷ SPLCSetDO(7, 1)		.8 🔵 — 🗢	39.719		DO0 DO4	DO1 DO5	DG2 DG6	DO3 DO1	
		t1⊷ WaitMs(20)		Profix			000	C01 C05	C02 C06	000	
		12- SPLCSatDO(7, 0)		Fornt reme	Add		Dia	DIS	012	DIS DI7	
		10⊷ SetDO(11,1,0,0)		Sensor	~		C10 C14	CH	CIE CIE	C0	

Obrázek 4.7-6-9 Příkazové rozhraní režimu

# 1.3.5.7.6.8 Příkaz ke kolizi

Kliknutím na ikonu "Collision" vstoupíte do rozhraní pro úpravu příkazu Collision.

Tento příkaz slouží k nastavení úrovně kolize. Prostřednictvím tohoto příkazu lze úroveň kolize každé osy nastavit v reálném čase během provozu programu a scénář aplikace lze nasadit pružněji.



Obrázek 4.7-6-10 Rozhraní příkazů pro kolize

# 1.3.5.7.6.9 Příkaz Acc

Kliknutím na ikonu "Acc" vstoupíte do rozhraní pro úpravu příkazů Acc.

Příkaz Acc slouží k realizaci funkce, kterou lze samostatně nastavit zrychlení robota. Nastavením faktoru škálování zrychlení pohybového příkazu lze zvýšit nebo snížit dobu zrychlení a zpomalení a upravit dobu taktu akce robota.



Obrázek 4.7-6-11 Příkazové rozhraní Acc

# 1.3.5.7.7 Periferní příkazové rozhraní



Obrázek 4.7-7 Periferní příkazové rozhraní

# 1.3.5.7.7.1 Příkaz Gripper

Kliknutím na ikonu "Gripper" vstoupíte do rozhraní pro úpravu příkazů Gripper.

Tento příkaz se dělí na příkaz pro řízení pohybu chapadla a příkaz pro aktivaci/reset chapadla. V příkazu pro ovládání chapadla se zobrazuje číslo chapadla, které bylo nakonfigurováno a aktivováno. Uživatel může provést úpravy prostřednictvím editačního pole nebo posunout posuvník na Požadovaná hodnota slouží k dokončení nastavení otevírání a zavírání čelistí, rychlosti otevírání a zavírání a momentu otevírání a zavírání. Blokování znamená, že se chapadlo pohybuje paralelně s předchozím pohybovým příkazem. Kliknutím na tlačítka "Přidat" a "Použít" uložte nastavenou hodnotu do výukového souboru. Příkaz resetování/aktivace chapadla zobrazí počet nastavených chapadel a příkaz resetování/aktivace lze přidat do programu.

ā =	C.	$\Theta $	ii)				s	topped toolco	oord0 wobj0 exaxis	0 10		<u> </u>	12 8
		000			Gripper		×				- 11		
initialize <	📹 🖻 🛎		Jaw No	1 .	-			Operation&Stat	as 360" Free Mounting F	Fixed Maun	ing		
📄 Teaching 🗸	Peripheral com ~	Process1A.lua	Gripper	0	0	•	100	Move Car		Robot i	Pose		(III)
Program Teachi	command	1++	Opening	0	0	•	100	RCM			605 (N-1)	10 #10 10	Joints
	Gapper Spray	2++ SetWOb)Coord(1,464.500,820.000	and closing	<u>.</u>				**************************************		J4 1-107	864 .15 : 4	51.091 JG	: 39.719
	1°1 🕾	3⊷ SetToolCoord(2,-2.500,0.000,644	speed	(ñ	0		100			X 1090	102 V 14	71.809 Z	TCP
	Eaxis Convey	4→ MoveJ( 0.670, -55.310, 134.370, -1	and closing		1.3				-	RX:-151	395 HY: 7	2.646 HZ	100.009
		5++ SPLCSetDO(1,1)	torque	-	1			-138.896		Tota			FT
💥 Settings		0++ SPLCSetAO(0,10)	time	0	ms			-79.428		Fx :0.00 Tx :0.00	) Fy:0 Ty:0	.000 Fz .000 Tz	10.000
		7⊷ SPLCSetAO(1,0)	Whether it is blocked	block	v			75	9	Act_State	£1		an area
		B⊷ WaitDi (1,1,0,2)						-107 864		-			Line-Num
		9⊷+ SPLCSetDO(1,0)					Add	-91.891		Not 1	10.0.9		Children
		10⊷ SPLCSetDO(7, 1)	-			()		39.719		DOG	DO1	D02	DD3
		t1⊷ WaitMs(20)	Jaw No	0		1	~			000 CO4	001	C02 C06	C03 E07
		12 SPLCSetDO(7, 0)	Reset				Active	Add		Dia	Dit	012	DIS DI7
		10⊷ SetDO(11,1,0,0)	Added Comr	nands				~		C10 C14	CCH	CI2 CIE	CI0 CI7

Obrázek 4.7-7-1 Příkazové rozhraní chapadla

# 1.3.5.7.7.2 Příkaz Spray

Kliknutím na ikonu "Spray" vstoupíte do rozhraní pro úpravu příkazu Spray.

Tento příkaz je příkazem souvisejícím se stříkáním, který ovládá stříkací pistoli, aby "spustila stříkání", "zastavila stříkání", "spustila čištění pistole" a "zastavila světelnou pistoli". Při úpravě programového příkazu je nutné potvrdit, že byly nakonfigurovány periferie stříkací pistole. Podrobnosti naleznete v kapitole o periferiích robota.
⊴ ≡			IJ	s	toolcoord0	wobj0 exaxis0	10			12 8
@ initialize <	📹 🖻 🛎		Spray	,	Operation&Status 3	10" Free Mounting Fix	ad Mauni	ing		
📄 Teaching 🗸	Peripheral com ~	Process1A.lua	Start spraying	Add	Nove Ca.		Robot P	ose		) (±)
Program Teachi	command	3↔	Stop spraying	Add	RCM		J) 1-118.0	195 - 12 : - 1	P9 458 .19	Joints 175
Graphical Program	Gripper Bpray	2++ SetWOb)Coord(1,464.500,820.000,	Clear gun	Add	2/9/2		J4 (+107.)	64 .16 : 4	JL 1001 JD	: 39.719
Manage Teaching	1°1 🖂	3↔ SetToolCoord(2,-2.600,0.000,644.0	Stop clearing		- (* ·		X 1390.0	21 V 14	71.009 Z	TCP
4. Status <	Eaxis Convery	4↔ MoveJ( 0.670, -55.310, 134.370, -16		Add		-	RX:-151	195 FEY: 7	2.646 RZ	1 -86.909
🔠 Auxiliary <		5+++ SPLCSetDO(1,1)	Added Commands.		-138.896	-	Tota			FT
🛠 Settings		0++ SPLCSetAO(0,10)			-79.428		Fx :0.000	Fy 10 Ty 10	000 Fz	10.000
		7↔ SPLCSetAO(1,0)		Apply	75	6	Act_State	1		
		B⊷ WaitDI (1,1,0,2)		.34 🔵 — O — 🤇	-107 864	10	in the second se			Line-Num
		9⊷ SPLCSetDO(1,0)		J5 🔵 -0 — (	-91.891		Nort : D	0.0.0		
		10+++ SPLC SetDO(7, 1)		.s 🔵 —o- (	39.719		000	DOI	DO2	CtriBex DD3
		t1⊷ WaitMs(20)		Profix			004 004	C01 C05	C02	C03
		12⊷ SPLCSetDO(7, 0)		Foint same	Add		Dia	Dis	012	Dia
		10⊷ SetDO(11,1,0,0)		Sensor	~		C10 C14	CCH	CI2 CIE	CID

Obrázek 4.7-7-2 Rozhraní příkazu Spray

### 1.3.5.7.7.3 Příkaz EAxis

Kliknutím na ikonu "EAxis" vstoupíte do rozhraní pro úpravu příkazů EAxis.

Tento příkaz se používá v kombinaci s příkazem PTP pro scénáře využívající externí osy a může rozložit pohyb bodu v prostoru ve směru osy X na pohyb externích os. Vyberte číslo externí osy, zvolte režim synchronního pohybu, vyberte bod, kterého má být dosaženo, a kliknutím na tlačítko "Přidat" a "Použít" příkaz uložte.

ā	=		I I I I I I I I I I I I I I I I I I I	EAxis	1	× opp	ed toolcoord	10 wobj0 exaxis	0   10			12 8
Initialize	< 1	1 B ±	2 8 8   9 × 9	communication configuration	-	Op	eration&Status	360" Free Mounting F	ixed Mount	Ing		
Teaching	~ P	eripheral com ~	C Process1A.lua	Port number		I Mo			Robot P	ose		æ
Program Tea	chi	semmand	1↔	Communication cycle	1					195 .12	79.458	Joints
Graphical Pro	gtam g	Sapper Bipary	2++ SetWOb)Coord(1,464,500,820.000	Loid	Configure		"/8°2		JA :-107	864 JB : 4	51.091 JG	1 29.719
Manage Teac	hing	Easte Conver	3↔ SetToolCoord(2,-2.500,0.000,644.	Motion command			•		X :390.3	00 V :4	71.005 Z	1 546 499
4. Status	<		1↔ MoveJ( 0.670, -55.310, 134.370, -1	Sports mode	asynchronous ~	9			NALTISC.		C 040 PZ	
88 Auxiliary	<		5++ SPLCSetDO(1,1)	Point Name:	1 ~		-1.38.890		Tota			
🛠 Settings			6⊷ SPLCSetAO(0,10)	Tool coordinate system:	hiest		75		Tx :0.000	Ty 10	000 Tz	10.000
			7⊷ SPLCSetAO(1,0)	Workpiece coordinate system	0		-107.864	9				Line, Num
			S⊷ Wattbi (1,1,0,2)	E1	0.000	bi	-91.891		Not : D	0.0.0		
			9++ SPLCSetDO(1,0)	E2	0.000		39.718		DOD	0.01		CtriBox
			the Weithle/201	E3	0.000				D04 000	D05 001	006	D01 C03
			12m SPLC8#00(7.0)	E4	0.000		Add		CO4	005	00	ED7
			10 SetDO(11,1,0,0)	Commissioning speed	100 9				D14 C10	DIS	CI2	D/7 C/0
					Add				Cia	CIE	C.16	0.00

Obrázek 4.7-7-3 Příkazové rozhraní EAxis

### 1.3.5.7.7.4 Příkaz Convey

Kliknutím na ikonu "Convey" vstoupíte do rozhraní pro úpravu příkazů Convey.

Tento příkaz obsahuje čtyři příkazy: detekce polohy v reálném čase, detekce IO v reálném čase, zapnutí sledování a vypnutí sledování. Podrobnosti naleznete v kapitole Periferie robota.



Obrázek 4.7-7-4 Příkazové rozhraní dopravníku

# 1.3.5.7.8 Rozhraní příkazů pro svařování



Obrázek 4.7-8 Rozhraní příkazů pro svařování

# 1.3.5.7.8.1 Příkaz ke svařování

Kliknutím na ikonu "Weld" vstoupíte do rozhraní pro úpravu příkazu Weld.

Tento příkaz se používá hlavně pro periferie svařovacího stroje. Před přidáním tohoto příkazu se ujistěte, zda je v uživatelských periferiích dokončena konfigurace svářečky. Podrobnosti naleznete v kapitole o periferiích robota.

ā ≡		I		Stopped toolcoord0	wobj0 exaxis0 1	• • • •		12 8
🐵 initialize 🧹 📹 🖪 d	5 <b>8 8</b> 8 7 8 7 8	Wel	d d	× Operation&Status 380	Free Mounting Fixed h	sunting		
Teaching Velding commi-	Process1A.lua	Welding process No	Q		Ro	bot Pose		-
Program Teachl 🜱 📝	1++	Maximum waiting time	10000	MS		.118.807 12 1	70.458	Joints
Graphical Program Weld Segned	2⊷ SetWOb)Coord(1,464.500,820.000,	Are	End a	re	ж	-107 864 .15 : -	51.091 .0	: 99.719
Manage Teaching	3⊷ SetToolCoord(2,-2.500,0.000,644.0	Added Commands				355.30E V :-	471 509 2	TCP
14. Status 🦿 🏹 🕅	4↔ MoveJ( 0.670, -55.310, 134.370, -10	6			RX	-161 395 HTT: 1	72.646 92	809.86-12
B Auxiliary C Weierch WeierTh	5+++ SPLCSetDO(1,1)		Appl	-138.897	77.			FT
🛠 Settings 💦	6⊷ SPLCSetAD(0,10)			-79.428	Fx	0.000 Ey 10	1.000 Fz	10.000
	7⊷ SPLCSetAO(1,0)		72 👄 — O	75	Act	State: 1		
	B⊷ WaitDi (1,1,0,2)		.J4 🔵 —O-	107 864	6			Line-Num
	9++ SPLCSetDO(1,0)		J5 🔵 -O	- 🕀 [-91.891				
	10++ SPLC SetDO(7, 1)		.6 🔵 —0	39.719		00 001	DO2	DD3
	t1⊷ WaitMs(20)		Profix			04 C05	006	001
	12 SPLC SetDO(7, 0)		Point same	Add	OF	Ditt	002	DIS
	10⊷ SetDO(11,1,0,0)		Sensor	~	0		CI5 CI5	CID

Obrázek 4.7-8-1 Rozhraní příkazů pro svařování

# 1.3.5.7.8.2 Příkaz Segment

Kliknutím na ikonu "Segment" vstoupíte do rozhraní pro úpravu příkazů segmentu.

Tento pokyn je speciálním pokynem pro svařování, který se používá hlavně v případě přerušovaného svařování, kdy se jeden úsek svařuje a jeden úsek se nesvařuje. Mezi počátečním a koncovým bodem použijte tento příkaz k výběru počátečního a koncového bodu, nastavení rychlosti ladění, nastavení portu DO pro start oblouku, délky provedení, délky neprovedení, nastavení režimu funkce podle aktuální scény použití, volby kyvů a pravidel zaokrouhlování. Realizujte funkci segmentového svařování.

ā	=		<u>ن</u> ا ک	I	S	topped toolcoord0 wobj0 exa	axis0   10   🖢   🕰   🔞   🗺   😒
۲	initialize 🤇	<b>₩ 6</b> ±		Segm	ent X	Operation&Status 360" Free Mountin	g Fixed Mounting
	Teaching 💛	Welding command	Process1A.lua	starting point	333 🗸		Robot Pose
Pr	ogram Teachi	9 🗉	1++	End	A1start ~		Joints
		Weld Segment	2⊷ SetWOb)Coord(1,464.500,820.000	Commissioning speed	100 %	*/902	J4 :-107.864 J6 :-51.891 J6 : 39.719
16		Laser LT-Rec	3↔ SetToolCoord(2,-2.500,0.000,644		(margaret 1)		TCP X 1560 300 V 1471 505 2 1546 496
		🎾 💟	d↔ MoveJ( 0.670, -55.310, 134.370, -1	Arc starting do port	Ctrl-DO6 ~		RX:-151 395 RY: 12.646 RZ:-66.908
88		Weiearch Weid-Tic	5+++ SPLCSetDO(1,1)	Non executive length	mm	-138.890	FT Tor
*		Adjust	0⊷ SPLCSetAO(0,10)	Functional mode	First section over w	-79.428	Fx:0.000 Fy:0.000 Fz:0.000 Tx:0.000 Ty:0.000 Tz:0.000
		122.00	7++ SPLCSetAO(1,0)	Swing selection	The actuation sac v	75	Act_State: 1
			8⊷ WaitDi (1,1,0,2)	Rounding Rules	No rounding ~	-107 864	Line-Num
			9⊷+ SPLCSetDO(1,0)			91.891	ANT . LOLO
			10++ SPLCSetDO(7, 1)		Add	39.719	D00 D01 D02 D03
			t1⊷ WaitMs(20)	Added Commands			CON COS COS COS
			12 SPLCSetDO(7, 0)			Add	Dia Dri Dia Dia
			13⊷ SetDO(11,1,0,0)		Apply		CIO CH CI2 CO CI4 CI5 CI6 CI7

Obrázek 4.7-8-2 Příkazové rozhraní segmentu

#### 1.3.5.7.8.3 Laserový příkaz

Kliknutím na ikonu "Laser" vstoupíte do rozhraní pro úpravu příkazů laseru.

Tento příkaz se skládá ze tří částí: příkaz pro laser, příkaz pro sledování a příkaz pro polohování. Před přidáním tohoto příkazu zkontrolujte, zda byl laserový sledovací senzor v uživatelské periferii úspěšně nakonfigurován. Podrobnosti naleznete v kapitole Periferie robota.

(±)
Joints 71.12. 828.07.1
i:-51.091 J5:39.719
TCP : 471.009 Z : 546.498
Y: 72.646 RZ: -66.909
10.000 Te : 0.000
Line-Num
CtriBox
6 006 007 11 002 009
CI2 CI2 CI2 CI0 CI5 CI7

Obrázek 4.7-8-3 Příkazové rozhraní laseru

### 1.3.5.7.8.4 Příkaz LT-Rec

Kliknutím na ikonu "LT-Rec" vstoupíte do rozhraní pro úpravu příkazů LT-Rec.

Tento příkaz realizuje funkci vyjmutí počátečního a koncového bodu záznamu laserového sledování, takže robot se může automaticky přesunout do polohy počátečního bodu, což je vhodné pro případ, kdy pohyb začíná M om vnější strany obrobku a provádí se záznam laserového sledování. Současně může hostitelský počítač získat informace o počátečním a koncovém bodě v zaznamenaných datech. pro následné cvičení.

Realizujte nastavitelnou funkci laserového sledování a rychlosti reprodukce, takže robot může zaznamenávat velmi vysokou rychlostí a poté reprodukovat podle běžné rychlosti svařování, což může zvýšit efektivitu práce.



Obrázek 4.7-8-4 Příkazové rozhraní LT-Rec

### 1.3.5.7.8.5 Příkaz W-Search

Kliknutím na ikonu "W-Search" vstoupíte do rozhraní pro úpravu příkazů W-Search.

Tento příkaz je příkazem pro polohování svařovacího drátu, který obsahuje tři příkazy: začátek polohování, konec polohování a výpočet posunu. Tento příkaz se obvykle používá ve svařovacích scénách a vyžaduje kombinaci příkazů IO a pohybu svařovacího stroje a robota.

ā	=	с.	<b>G D</b>	U	s	Stopped toolcoord0 wobj0 exaxis	s0 10 <b>6 A 8 12</b> 8
۲	initialize 🔇	± ۵ 📽		WiSearch	×	Operation&Status 360" Free Mounting 1	Fixed Mounting
	Teaching 🗸	Welding commi~	Process1A.lua	Welding wire locating setting		More (d)	Robot Pose
Pr	ogram Teachi	9 🗐	1++	Datum position	Do not update 🗸 🗸	FIGH -	Joints
G		Weld Segment	2++ SetWOb)Coord(1,464.500,820.00	Locating speed	10 %	*/502	J4 :-107.864 J6 : 51.091 J6 : 29.719
M		Laser LT.Rec	3⊷ SetToolCoord(2,-2.600,0.000,644	Locating distance	10 mm		TCP
4		3	d↔ MoveJ( 0.670, -55.310, 134.370, -	Automatic return fiag	Do not return eutr~		X 1990.300 V 1471.509 2 1546.496 RX1-161.395 RY172.646 RZ1-66.309
88		Williamon Weld-Tic	5++ SPLCSetDO(1,1)	Automatic return speed	10 %	-138.890	FT FT
*		Adust	6++ SPLCSetAO(0,10)	Automatic return distance	10 mm	D .70 428	Tour Fx:0.000 Ey:0.000 Fz:0.000 Tx:0.000 Tx:0.000 Fr:0.000
		707A3A21	7↔ SPLCSetAO(1,0)	Locating mode	Positioning of tea	D 75	Act_State: 1
			B⊷ WaitDi (1,1,0,2)	End locating	Start locating	-107 864	Line-Num
			B++ SPLCSetDO(1,0)			91.891	Note : E00.0
			10+++ SPLCSetDO(7, 1)	Weld type	Fillet weld	39.719	CtriBex
			t1⊷ WaltMs(20)	computing method	1D (one of xyz) ~		CO4 CO5 CO6 E07
			12 SPLCSetDO(7, 0)	Add point b	Add point a	Add	
			10⊷ SetDO(11,1,0,0)	1		<b>~</b>	Ci0 CH CI2 CB Ci4 Ci5 Ci6 CI7

Obrázek 4.7-8-5 Příkazové rozhraní W-Search

Při psaní programu se obvykle nejprve nastaví příkaz start hledání a poté se přidají dvě instrukce LIN, které určují směr hledání. Po úspěšném vyhledávání získáme vypočtený offset a ten předáme příkazem overall offset. Chcete-li se projevit do skutečného příkazu pohybu svařování, je příklad programu následující.



Obrázek 4.7-8-6 Příklad vyhledávání W(1D)

# 1.3.5.7.8.6 Příkaz Weld-Trc

Kliknutím na ikonu "Weld-Trc" vstoupíte do rozhraní pro editaci příkazů Weld-Trc.

Tento příkaz realizuje sledování švu robota a využívá detekci odchylky svařovacího švu ke kompenzaci trajektorie, přičemž k detekci odchylky švu lze použít senzor oblouku.

	-		Weld-Trc	×		
	📹 🖬 🖆				Operation&Status 360" Free M	Aganting Fixed Mounting
	Weldon common		Compensation for upp	×		
E Teaching 🗸	Welding command	D 0815.ks	Left and right deviation	v	1 Hour Car	Robot Pose
Program Teachin	• 7 F	1↔	Left and right adjustme			Jain
	Weld Segment	2++ pos = ()	Up and down adjustme		96	J4 :-110.745 J6 :-82.967 J6 :-30.119
	Laser LT-Rec	3⊷ pos = GetWeldTrackingRecordSt	Left and right start com	cyc		TO
	🌮 👔	$4 \leftrightarrow \text{if type(pos)} == "table" then$	Up and down start com	сус	27	RX:175.073 RY:29.542 RZ:04.014
🗄 Auxiliary <	Webearch Weld-Tic	5++ laserPTP(#pos,pos)	Maximum compensatio	mm	41.429	Torr.
🗙 Settings	Actions	G⇔+ end	Maximum compensatio	mm	95 544	Fx:0.000 Fy:0.000 F2:0.000 Tx:0.000 Ty:0.000 Tr:0.000
	107ALALI	7↔ Laser3ensorRecord(0,10,30)	Total left and right maxi	mm	129 318	Act_State; 1
			Maximum compensatio	mm	-110.745	Line-Nu
			Upper and lower refere	mA	62 987	Nat : 000.9
			Selection of upper and	~	-20 119	Dot Dot Doz DOJ
			Setting method of uppe	~		004 005 006 007 000 001 002 003
			ure tracking of	tre tracking or	Add	DIA DI1 DI2 DI3
				and the second second		Di4 DIS DIR DI7
			Added Commands		~	CCI0 CCH CCI2 CCI0

Obrázek 4.7-8-7 Příkazové rozhraní Weld-Trc

# 1.3.5.7.8.7 Příkaz Adjust

Kliknutím na ikonu "Adjust" vstoupíte do rozhraní pro úpravu příkazu Adjust.

Tento příkaz adaptivně upravuje polohu svařovacího hořáku pro scénu sledování svařování. Po zaznamenání tří odpovídajících bodů polohy přidejte příkaz adaptivního nastavení polohy podle skutečného směru pohybu robota. Podrobnosti naleznete v kapitole Periferie robota.

• •	5	Stooped tool	lcoord0 wobj0 exaxis0	10		8 12 8
1 9 7	(CS)	veration&St	itatus 360° Free Mounting Fi	and Mounting	Q	
Sheet type	Corrugated plate	•		Robot Pos	ie i	(±
Direction of movement	Left to right	•				Joints
Attitude adjustment time ord(1.464.500.820.0	1000	ms 💊		J1 :-138.899 J4 :-107.964	8 .12 : -79.438 1 JE : -91.891	.13 : 75 .36 : 99.719
Length of the first	100	mm */9^2				TCP
70, -55,310, 134,370, inflection point type	From top to botto	•		X :090.302 RX:-161.390	Y :471.000 9 RY112.040	Z : 546.409 RZ: -06.900
Length of the second	100	mm -138.896				FT
section		-79.428		Tool Fx : 0.000	Fy:0.000	Fz:0.000
Length of the third S(1,0) segment	100	mm 75		Tx : 0.000 Act_State: 1	ту : 0:000	72:0.000
0,2) Length of the fourth	100	mm -107.864	1	-		Line-Num
D(1,0) Length of the fifth	300	-91.891		Num : 0.0.0	0.0	
iO(7, 1) segment	100	39.719	•	DOG	DO1 00	CtriBox
) Point a rx: 0.000	ry 0.000 rz 0.000			004 CO6	005 0	06 007
Point b nc 0.000	ry: 0.000 rz: 0.000	Add		CD10 0	DH OD	R2 C DIS
Point c rx: 0.000	ry: 0.000 rz: 0.000			Ci0	CH C	16 D17 32 C13
	Image: Second system       Add         Image: Second system       Sheet type         ua       Direction of movement         Attitude adjustment time       Attitude adjustment time         var(2,-2.500,0.00,44       Length of the first section         var(2,-2.500,0.00,44       section         var(2,-2.500,0.00,44       section         var(2,-2.500,0.00,44       section         var(1,1)       Length of the first second section         var(1,1)       Length of the second section         var(1,0)       Length of the fourth segment         var(1,0)       Length of the fifth segment         var(1,0)       Point a nx: 0.000         var(2,0)       Point b nx: 0.000         var(2,0)       Point c nx: 0.000	Adjust         Adjust         Adjust         Sheet type       Corrugated plate         Direction of movement       Left lo right         Attriude adjustment time       1000         ord(1,464.500.820.0)       Length of the first       100         vd(2,2.5.00,8.00.6,64       Section       100         resction       Inflection point type       From top to botto         vd(1,1)       Length of the second       100         vd(1,0)       Length of the fourth       100         vd(1,0)       Length of the fourth       100         vd(1,0)       Length of the fourth       100         vd(1,0)       Length of the fifth       100         vd(7, 1)       Point a rx 0.000 ry 0.000 rz 0.000         vd(7, 0)       Point b rx 0.000 ry 0.000 rz 0.000         vd(7, 0)       Point c rx 0.000 ry 0.000 rz 0.000	Stopped         Stopped         Ioo           Adjust         erations is           Sheet type         Corrugated plate 1         erations is           Direction of movement         Left lo right 1         erations is           Attitude adjustment time         1000         ms         ***/2           ard(2,2,8600,800,666         Length of the first section         100         mm         *38.896           70, -56,310, 134.370,         Inflection point type         From top to both 1         -7.9428         -7.9428           20,10)         Length of the second section         100         mm         -138.896           20,10)         Length of the fourth segment         100         mm         -137.864           20,10)         Length of the fourth segment         100         mm         -137.864           21,0)         Length of the fourth segment         100         mm         -137.864           21,0)         Length of the fourth segment         100         mm         -137.864           21,0)         Length of the fourth segment         100         mm         -137.864           20,7,1)         Point a n:: 0.000         ny: 0.000         rz: 0.000         Point 3           20,0)         Point b n:: 0.000         ny:	New Year       Stopped       toolcoord0       wobj0       exaxist         Adjust       Adjust       erration53tate       \$80° Free Mounting       Fill         ua       Direction of movement       Left to right       •       •       •         Attitude adjustment time       1000       ms       •       •       •         rd(1,464.500.820.0)       Length of the first       100       mm       •       •       •         rd(2,2.800.0.006.46       Section       100       mm       •       •       •       •         rd(1,464.500.820.0)       Length of the first       100       mm       •	Stopped         toolcoord0         wobj0         exaxis0         10           Adjust         areadon&Status         300° Free Mounting         Freed Mounting           Sheet type         Corrugated plate *         Robot Pool           Direction of movement         Left to right *         Robot Pool           Attitude adjustment time         1000         mm         79/2           rrd(2,2,800,0006,64         Length of the first second section         100         mm         79/2           Attitude adjustment time         1000         mm         79/2         700°         70°         700°	Note:         Stopped         toolcoord0         wobj0         exaction 0         wobj0



#### 1.3.5.7.9 Rozhraní příkazů Force Control

Obrázek 4.7-9 Rozhraní příkazů Force control

### 1.3.5.7.9.1 Příkaz F/T

Kliknutím na ikonu "F/T" vstoupíte do rozhraní pro úpravu příkazů F/T.

Instrukce zahrnují FT\_Guard (detekce kolize), FT\_Control (řízení konstantní silou), FT\_Spiral (spirálové vkládání), FT\_Rot (rotační vkládání), FT\_Lin (lineární vkládání), FT\_FindSurface (polohování povrchu), FT\_CalCenter (polohování na střed) sedm instrukcí, podrobně viz kapitola Periferie robota.



Obrázek 4.7-9-1 Příkazové rozhraní F/T

### 1.3.5.7.9.2 Příkaz krouticího momentu

Kliknutím na ikonu "Torque" vstoupíte do rozhraní pro úpravu příkazů Torque.

Tento příkaz je příkazem pro záznam točivého momentu, který realizuje funkci detekce kolize v reálném čase. Kliknutím na tlačítko "Torque Record Start" (Spuštění záznamu momentu) se nepřetržitě zaznamenává kolizní situace během činnosti příkazu pohybu a zaznamenaný moment v reálném čase se používá jako teoretická hodnota pro posouzení detekce kolize, aby se snížila pravděpodobnost falešných poplachů. Při překročení nastaveného prahového rozsahu se zaznamená doba trvání detekce kolize. Klepnutím na tlačítko "Torque Recording Stop" (Zastavení záznamu krouticího momentu) zastavíte záznam. Kliknutím na tlačítko "Torque Record Reset" (Obnovit záznam točivého momentu) vrátíte stav do výchozího stavu.

⊴ ≡			II				s	topped toolcoord0 v	vobj0 exaxis	0   10	- 4		12 8
initialize <	<b>4 6</b> 4				Torque		×	Operation&Status 360" F	Free Mounting	ixed Maun	gnt		
🔲 Teaching 🗸	Force Control C~	🗅 323.lua	Smoot	h Selection	2	No smooth	ing (n 🗸	Nort 🚠		Robot F	ose		æ
Program Teachi Graphical Program	Command	<pre>1 MireSearchStart(0,10,10,10,0,10 2 Lin(1,100,-1,0,0) 3 Lin(1,100,-1,0,0)</pre>	Joint	Negative threshold	Positive (Nm)threshold	Collision detection (Nm)duration	n	*19*2		ut 1-198 .04 1-107	897 .12 : -7 864 .16 : -5	9.438 .13 1.091 .16	Joints 175 199.719
Manage Teaching		4 WireSearchEnd(0,10,10,0,10,	J1	-0.1	0.1	500	ms			X 1390.1 RX:-161	IOE Y :4	71.009 Z 2.646 RZ	TCP 1 545 499 1 -85 909
H Auxiliary <			J2 J3	-0.1	0.1	500	ms	-138.897	-				FT
🗙 Settings			J4	-0.1	0.1	500	ms	.79.428		Tour Fx : 0.000 Tx : 0.000	) ≓y×0 †y±0	000 Fz 000 Tz	: 0.000 : 0.000
			JS	-0.1	0.1	500	ms		1	Act_State	:1		
			16	-0.1	0.1	500	ms	-91.891		Nutt 1 1	0.0.0		Line-Num
							stop Teset	39.719		D00 D04 C00	D01 D06 C01	002 006 002	CtriBex DD3 DO7 CO3
			Added	Commands				Add		CO4 Dia Dia Ci0 Ci4	DH DIS CH CIS	C06 012 018 C12 C15	007 DIS DI7 CI7

Obrázek 4.7-9-2 Rozhraní příkazu Torque

# 1.3.5.7.10 Vizuální příkazové rozhraní

≂ ⊼		Stopped toolcoord0 wobj0 exaxis0 10 🖢 🛦 🕅	
🔿 Initalize 🧹 🗂 🗟 🛛	┶┶■■↓♥⊁≋歯≙⊽│≓	Coperation&Status 360° Free Mounting Fixed Mounting	
Initialize  Teaching  Visual comman Visual common Visu		Control         Control <t< th=""><th>Joints     Joints     Joints</th></t<>	Joints     Joints

Obrázek 4.7-10 Vizuální příkazové rozhraní

### 1.3.5.7.10.1 3D příkaz

Kliknutím na ikonu "3D" vstoupíte do rozhraní pro úpravu 3D příkazů.

Tento příkaz generuje příkazy pro příklady programů 3D vidění. Uživatelé se mohou na vygenerované programy odkazovat a komunikovat s jinými zařízeními pro vidění, včetně dvou referenčních případů programů kalibrace kamery a snímání kamery.



Obrázek 4.7-10-1 Příkazové rozhraní 3D

# 1.3.5.7.11 Příkazové rozhraní pro paletizaci



Obrázek 4.7-11 Příkazové rozhraní pro paletizaci

# 1.3.5.7.11.1 Příkaz pro paletu

Kliknutím na ikonu "Paleta" vstoupíte do rozhraní pro úpravu příkazu Paleta.

Tato instrukce generuje instrukce pro paletizační program, který je v souladu s funkcí pohybu matic v části 3.9.6. Podrobnosti naleznete v této kapitole.

ā	-	÷		P	allet	× opped toolcoord0	wobi0 exaxis0	10 1		ছি	BA
0	Initialize <	<b>≝ 6</b> ±	7 8 8   9 × 1	Manipulator motion mode Motion selection	e 🗸	Operation&Status 36	T Free Mounting Fixe	ed Mauril	ing		
	Teaching 🗸	Palletizing instr ~	Process1A.lua	Polyst motion with		Allow Ca.		Robot P	ose		æ
P	rogram Teachi	instructions	1++	Path selection	Head to tail walkir~	RGM			197 .12 . 78	er. sea e	Joints
G		Pallet	2⊷ SetWOb)Coord(1,464.590,820.000	0		*75°2		.34 :-107.8	i64 .16 : -5	at. 160.Fi	: 39.719
M			3↔ SetToolCoord(2,-2.500,0.000,644,	-	\$P	<b>_</b> *		N 1350 N	a v sa	71 509 2	TCP
4			i↔ MoveJį 0.670, -55.310, 134.370, -1	Stacking mode satting		-	-	RX:-161.2	195 R.Y. 1	2.646 HZ	1 -86.909
88			5+++ SPLCSetDO(1,1)	Mode selection	Stacking	-138.897	-	Tour			FT
*			0++ SPLCSetAO(0,10)	t,	•	-79.428	<u> </u>	Fx :0.000 Tx :0.000	Fy 10 Ty 10	000 Fz 000 Tz	1 0 000 1 0 000
			7↔ SPLCSetAO(1,0)	L	*	75	8	Act_State:	1		
			B⊷ WaitDl (1,1,0,2)			-107 864	76	Marrie D	000		Line-Num
			0+++ SPLCSetDO(1,0)	First point	g to the path	91.891					Chillen
			10+++ SPLC SetDO(7, 1)	Second point	1 ~	39.719		DOG	DOT	DO2	DOS
			t1⊷ WaltMs(20)	Third point	1 ~			000 004	C01 C05	C02	001
			12 SPLCSetDO(7, 0)		3 <del>1</del>	Add		DIO	DH	012	DIS
			10⊷ SetDO(11,1,0,0)	Number of rows and colu	mns			C10 C14	CIT	CI2 CIE	CIP
						-					

Obrázek 4.7-11-1 Příkazové rozhraní palety

# 1.3.5.7.12 Komunikační příkazové rozhraní



Obrázek 4.7-12 Komunikační příkazové rozhraní

# 1.3.5.7.12.1 Příkaz Modbus

Kliknutím na ikonu "Mobus" vstoupíte do rozhraní pro editaci příkazů Modbus.

Příkazová funkce je sběrnicová funkce založená na protokolu ModbusTCP. Uživatel může ovládat robota a komunikovat s klientem nebo serverem ModbusTCP (stanice Master komunikuje se stanicí Slave) prostřednictvím příslušných instrukcí a provádět operace čtení a zápisu na cívkách, diskrétních veličinách a registrech.

Příklad čtení cívky modbus master



Příklad zapisovací cívky modbus master

	Stopped toolcoord0 wobj0 exaxis0 60 📿 🛦 👿 🌿 🛞
± ± □ □ □ □ ↓ □ ★ □ □ △ ⊽   ≓	Image: Slave - Mbslave1         File Edit Connection Setup Display View Window Help         Image: Im
2 - if err = 0 then 3 $data = \{1, 1, 0, 1, 1\}$	D Mbslave1 ID = 1: F = 01
Ht 4 modousHasterSetColls(10,0, modta, 0ata) 5 end 6 ModbusHasterClose(1)	Alias 00000 1 1 1

Příklad čtení a zápisu cívky Modbus slave

		(	() () (■) (□)		Runnir	ng toolcoord0	wobj0	exaxis0	60	C		8	8
-	ك ه	£		Ele Edit	ous Poll - Mbpoll1	etup Functions Di	splay <u>V</u> iew 15 16 22	Window H	elp 17				
Pause		1 e 2 * i 3	rr,id - ModbusSlaveCreate("0.0.0.0",502,1) :f err 0 then tab - (1.1.0.0.1)	Tx = 77	oll1 : Err = 47: ID = 1	1: F = 01: SR = 1000	lms		P.				
//eave	Segment	4	<pre>ModbusSlaveSetCoils(id,0,#tab,tab)</pre>		Alias	00000							
5	3	5 *	while 1 do	0	-	1							
Adjust	Gripper	6	<pre>data = ModbusSlaveGetCoils(id,0,5)</pre>	2		0							
13	r*1	7	a = data[1]	3		0							
Spray	EAxis	8	b = data[2]	4		1							
-0	1	9	c - dətə[3]	6		0							
Weld	Laser	10	d - data[4]	7		0							
=	m	11	e = data[5]	8		0							
Sonvey	F/T	12	RegisterVar("number","a")	19		0							
5	2	13	RegisterVar("number","b")										
3D	Pallet	14	RegisterVar("number","c")										
	20	15	RegisterVar("number","d")										
Offset	WSearch	16	RegisterVar("number","e")										
V	(V)	17 *	if a 0 then	Fas Hala									
Vir-IO	Thread	18	break;	ros neip. 1	hidaa Li*	-			a :	1.000	ь	: 1.000	5
V	V	19	end	传感器	点	汤加			c : e :	0.000 1.000	d	: 0.000	20
lensoC.	Modhus	20	end										

Pro více provozních funkcí ModbusTCP nás prosím kontaktujte pro konzultaci.

# 1.3.5.7.12.2 Příkaz Xmlrpc

Kliknutím na ikonu "Xmlroc" vstoupíte do rozhraní pro úpravu příkazů Xmlrpc.

XML-RPC je metoda vzdáleného volání procedur pro přenos dat mezi programy pomocí xml přes sokety. Tímto způsobem může řídicí jednotka robota zavolat funkci (s parametry) ve vzdáleném programu/službě a získat vrácená strukturovaná data. Řídicí jednotka robota se stará o všechny podrobnosti psaní zpráv klienta XML-RPC a o konverze mezi datovými typy a XML.



Obrázek 4.7-12-1 Rozhraní příkazu Xmlrpc

### Důležité:

- 1) Řadič funguje jako klient a připojuje se ke vzdálenému vlastnímu portu;
- 2) Řídicí jednotka funguje jako klient, který volá vzdálenou funkci;
- 3) Podpora volání různých vzdálených funkcí;
- 4) Podporuje vstup parametru pole řetězců a návrat výsledku pole znaků, počet prvků pole lze přizpůsobit;

Podporuje vstupní parametry pole dvojího typu a návrat výsledku pole dvojího typu, počet prvků pole lze přizpůsobit;

XmlrpcClientCall(serverUrl,methodName,tableType,param)

serverUrl url serveru, například "http://192.168.58.29:50000/RPC2"

methodName Název funkce volání, "example.add"

tableType 1-číselné pole, 2-řetězcové pole

param vol**á**ní parametr**ů** funkce

XmlrpcClientCall(error, result)

error 0 - žádná chyba, 1 - chyba

result Pokud je parametr předán jako dvojnásobné pole, výsledkem je dvojnásobné pole,

Pokud je parametr předán jako pole typu string, výsledkem bude pole...

↔typ**ř**etězce

# 1.3.5.7.13 Pomocné příkazové rozhraní



### Obrázek 4.7-13 Pomocné příkazové rozhraní

### 1.3.5.7.13.1 Příkaz vlákna

Kliknutím na ikonu "Vlákno" vstoupíte do rozhraní pro úpravu příkazu Vlákno.

Příkaz Thread je pomocná funkce vlákna. Uživatelé mohou definovat pomocné vlákno, které poběží současně s hlavním vláknem. Pomocné vlákno provádí především datovou interakci s externími zařízeními, podporuje komunikaci přes zásuvku, získává stav robota DI, nastavení stavu robota DO a získává informace o stavu robota. Datová interakce vlákna, data získaná hlavním vláknem prostřednictvím pomocného vlákna se používají k řízení úsudku logiky pohybu robota, snímek obrazovky příkladu uživatelského programu:



Obrázek 4.7-13-1 Příklad programu vlákna

# 1.3.5.7.13.2 Funkční příkaz

Kliknutím na ikonu "Funkce" vstoupíte do rozhraní pro úpravu příkazů funkce.

Tento příkaz má zavolat funkci rozhraní funkce, poskytnout zákazníkovi funkci rozhraní robota, aby si mohl vybrat, a připomenout zákazníkovi parametry požadované funkcí, což je pro zákazníka výhodné pro psaní příkazů skriptu, a další funkce se přidávají jedna po druhé.



Obrázek 4.7-13-2 Rozhraní funkčních příkazů

### 1.3.5.7.14 Kódování výukového programu

Výukový postup je rozdělen na stav šifrování a nešifrování. Stupeň šifrování se dělí na šifrování prvního stupně a sekundární šifrování. Mezi nimi je úroveň šifrování první úrovně nejvyšší a sekundární je druhá. Všechny výukové programy se zobrazují a nastavují ve formě informací o šifrování programu v "Nastavení systému - Vlastní informace". Popisy úrovní šifrování jsou uvedeny vpravo v tabulce.

> C O A	不安全 192.168.58.	2/index.html#/systemsetting				on le ☆ ★ □	4
<b>五座一下,如欧知道</b>	1 项目资料 🧧 threej	s 🧧 vue 🧧 lua 🧮 ace editor					Milets
र ≡				Stopped	toolcoord0 wobj0 exax	iso 60 🖌 🗛 🕲 ⊄	8
} Initializo <	9, General	Apply					
Teaching 🔇	Account	Demonstration of teaching procedures					
Status (	Plugin	Name Search					
Auxiliary <	< About	Program name	Not encrypted	First-level encryption	Second-level encryption	First-level encryption permission description	
Settings	Custom	congenzon.lua		600	(D)	Can read	
	<ul> <li>Marcain</li> </ul>	conyundom.lua			(1)	O Edit O Save	
		testzwb./ua		<b>6</b> D	<b>(</b> )	O Delete O Rename	
		bs.lua		(E)	(1)	Save As import	
		ptp.lua		<b>6</b>	<b>6</b> D	program Second-level coverage of	
		yyhuua		<b>6</b>	(iii)	encryption the same permission name	_
		yyhtest.kua		60	<b>6</b> D	description	- 1
		yyhtest1.lua		(ED)	•	Can read Save	- 1
		testscroll.lua	60		60	Save As Rename	- 1
	testscroll2.lua	<b>6</b>	<b>(</b> )		program coverage of		
						the same name	

Obrázek 4.7-55 Ukázka výukových postupů

Když je program ve stavu šifrování první úrovně, po otevření programu: odpovídající "export", "preser- vation", "existing as", "copy", "cut", "delete", "delete", "delete", "delete", "delete", "delete" Tlačítka jako "upward", "downward" a "editing mode switching" budou šedá. Klepněte na ikonu , která má být neplatná, a zobrazí se výzva, že se aktuální program je vzašifrovaném stavu. Ikona "přejmenovaného" programu se skryje. Přidávací lišty instrukcí a oblasti pro úpravu programu jsou neviditelné a vyzývají k uzamčení v první úrovni šifrování.



Obrázek 4.7-56 Rozhraní šifrování první úrovně programu

Když je program šifrování druhé úrovně, po otevření programu na stránce "Demonstrace programu": odpovídající "úspory", "kopírovat", "střih", "vložit", "vymazat", "horní", "horní" v ovládacím panelu Tlačítka jako "Přesunout" se změní na popel. Klikněte na ikonu, která má být neplatná, a zobrazí se výzva, že aktuální program je zašifrovaný. Ikona "přejmenovaného" programu se skryje. Přidávací lišta pokynů není viditelná a vyzve k uzamčení v sekundárním šifrování. V oblasti úprav programu lze normálně procházet čtecí program.

→ C 白 ▲ 不安全 192.168.5	8.2/index.html#/programteach		아 남 ☆ ★ 🛛 😩
五座一下, 地球知道 🧧 攻日35月 📒 the	ejs 🧧 vue 🔟 lua 🧮 aceleditor		8 Mile
? ≡		Stopped toolcoord0 wobj0 exax	iso   60   🖬   🕰   🗟   🛫   🛞
Initializo < 🚍 🗟 📩	☆■■↓☆★	Operation&Status 300" Free Mounting	Fixed Mounting
Teaching 🛩	Ctestscroli2.lua	Joint Base Tool Wold Move	Robot Pose
Program Teaching	1→ PTP(yyhtest1,100,-1,0)	Easts IO TPO FT RCM	Joints
Graphical Program	2 LoadTPD("yyhtest")	Speed 100 % Acceleration 180 "/s*2	J1 157 J2 -91249 J3 91193 J4 - 96327 J0 -91546 J0 - 45.06
Manage Teaching	3⊷ PTP(yyhtest14,100,-1.0)	Threshold 30 *	TCP
Status 🤇 🦳	s⇔ PTP(yyhtest1,100,-1,0)	Single Multi	RX: -172:604 RY: 4.558 RZ: 60.788
Auxiliary <	5++ PTP(yytitest14,100,-1,0)	JI 😄 — 🔶 15.7	FT
Settings	6⊷ PTP(yyhtest1,100,-1,0)	J2 😄 — • — 🛟 -91.249	Fx:0.000 Fy:0.000 Fz:0.000 Tx:0.000 Ty:0.000 Tz:0.000
encryption	7 PTP(yyhtest14,100,-1,0)	J3 😄 — 🗣 🔂 91.193	Act_State: 0
	8→ PTP(yyhtest1,100,-1.0)	J4 😄 — O — 🕀 -98.327	Line-Num
	S→ PTP(yyhtest14,100,-1.0)	J5 😑 🛟 -91.946	Chillen Chillen
	10 PTP(yyhtest1,100,-1.0)	J6 🔵 — 🔶 45.06	
	11- PTP(yyhtest14,100,-1,0)	Prets	CO0 CO1 CO2 CO3 CO4 CO5 CO5 CO5
	12 PTP(yyhtest1,100,-1,0)	Point name Add	

Obrázek 4.7-57 Rozhraní pro šifrování druhé úrovně programu

Funkci "export" lze použít jak pro šifrování první, tak pro šifrování druhé úrovně. Ověřovací operace budou

se při importu provede. Pokud je program stejného jména šifrovaným souborem, operace importu se přeruší a oznámí se, že pokrytí šifrovacího programu nelze zavést.

O FARE-INNOV × 8 HERREE ×	🗣 Google 📾 🛪 🛛 🕫 🐺 🛪 🖉 🍅	WebAPPRys × 0 FontAveson × 1 1 4 (7) (2013) × 0 FAIR-INNOV × G Collaboratio ×	+ ~ - 0 ×
← → ℃ ☆ ▲ 茶麦金   192.168.5	8.2/index.html#/programteach		* @ \$ <b>* 0 \$</b> 1
😵 五座一下,姚叔我回剧 🧧 项目资料 🧧 three	ejs 🧧 vue 🧧 lua 🥃 ace editor		🚺 其他书题
FR =		Stopped toolcoord0 webj0 exaxis0	60 🖬 🗛 🗐 🤕
- e militor 🖉 🖬 🗎 📥		Import File X Operation&Status 2007 Pres Reanting Pa	
🔲 Teaching 🗠	C testscroll2 lua	送择文件 testscroll.lua	Robot Pose
Program Teaching	1 #TP(yyhtest1,100,-1,0)		Jointa
Graphical Program	2 Loan TPD("yyhteet")	and the exception state, which assess the expected	11.157 22.41246 22.511日 月 - 48.227 月 - 41.049 次 - 41.06
Manage Teaching	3- PTP(yynes:14,100,-1,0)	Clave	TOP
an ann a' 🖉 🦱	< ₽TP(yyhtest1,100,<1,0)	Renato I	Roc -172 604 RY 4 528 02 65 788
	5 PTP(yymest14,100,-1.0)	Close Import	17
🗶 Satings 🛛 🚶 🛶 🕹	0 PTP(yyntest1,100,-1,0)		Tx: 0.200 Fx: 0.000 Fx: 0.100
Second-level encryption	7→ PTP(yyhtest14,100,-1.0)		Ad Bins (C
	3 PTP(yytdes11,100,-1,0)	# 🗢 — • — 🔂 [20.22]	Line-Hum
	0 #TP(yyntau114,100-1.0)	* <b>O</b> - <b>O</b> - <b>O</b> - <b>P</b> - <b>P</b> - <b>P</b>	Citillar
	10 FTP(yyttlesi1,100,-1.0)	* <b>—</b> — <b>—</b> • • • • • • • • • • • • • • • • • • •	1006 621 862 203
	11 FTF(yyntest14,100,-1,0)	Peets	Here Scot Man Here
🗄 testscroll2.lua 🧄 🔿			全部還示:X
24°C ∲⊼	Q 192 🧭 🖬	C 📮 🖴 🖸 🕸 💘 🗘 🤘 🔶 🤤 🖉 🗖 🗠 🔹	D O # @ O = 1554 O

Obrázek 4.7-58 Import programu

### 1.3.5.7.15 Místní výukový bod

Místní výukový bod je vázán na aktuální výukový program. Při přidání programového příkazu jej lze použít pouze pro aktuální výukový program a nelze jej použít v jiných výukových programech.

**Přidat**: Kliknutím na ikonu "Přidat místní výukový bod" vpravo od názvu souboru programu přidáte místní výukové body. (Podrobné záznamy o místních výukových bodech naleznete v záznamech výukových bodů v provozu robota).

a ≡		Stopped toolcoord0 wobj0 exakts	10 1 4 8 2 8
🕸 Initialize 🧹 🗖 🖪 🛛	± ± ■ ■   # ⊁ ∈ Ѣ ≙ ⊽   ≓	Operation& Status 200° Free Mounting Fixed Mounting	
E Teaching V All V	🗅 tast0724 lua	Joint Base Tool West Marve	Robot Pose
Program Teaching	1 ↔ PTP(q11,100,-1,8)	Cault 10 TPG FT RCM	Joints J1 : 43 772 J2 : -57 112 J3 : 76562
Graphical Program White II, Else	2↔ PTP(q22,100,-1,0)	Speed 100 % Acceleration 100 %012	J4 : -106.91 J5 : -36.245 J8 : -161.138
Manage Teaching Cots Wat	0++ #TP(q33,108,-1,5)	Trueshald 30 *	TCP 8266.663 Y - 497.654 Z - 191.134
A- Status < Ⅲ →	.4↔ PTP(q44,100,-1,0)	Single Matt	RXI-170.485 RYI-2.752 RZI-88.284
Bi Aunitary < Pause Defa	5++ PTP(q15,100,-1,0)	n 😄 —o— 🛟 🛛 🚈	FT
X Settings		22 🔵	Fx:0.000 Fy:0.000 Fz:0.101 Fx:0.000 Ty:0.000 T2:0.000
Motion command		J3 😄 — O— 🚯 75.552	Act_State: 1
STP UN		JA 😄	Line-Num
0.0		J5 😄 -0 🕀 -86245	10.2
ARC Orde		16 🔵 0 — 🌓 -161 139	000 001 002 003
A A			004 005 006 007 000 001 000 003
Spiral N-Spiral		Profix	004 005 006 007
22		Pointname Add	DIO CH1 D12 130 D14 D15 DW D17
Spline N-Spline		Sensor 👻	010 011 012 000 014 015 014 017
Stand TPD		Point name Add	Q A0201 E0% Q A0211 E2%
01 😁		Base coord OFF	<u>8</u>
Offset Served		Teni coord. OFF	EndEff DOD DO3
\$ \$		Workpiece ON	D10 D11
Trajactory Trajactory		Ext. axits c	Q Acad0: 10% Q Ard : 11%
HT HT			

Obrázek 4.7-59 Přidání místních výukových bodů

**Odstranit**: Kliknutím na sloupec s pořadovým číslem v tabulce vyberte místní výukový bod, který chcete odstranit, a poté klikněte na ikonu "Odstranit" v pravém horním rohu názvu místního výukového bodu, čímž místní výukový bod odstraníte.



Obrázek 4.7-60 Odstranění místního výukového bodu

**Spusťte**: Kliknutím na ikonu "Spustit spuštění" na panelu datových operací tabulky lokálního výukového bodu provedete jednobodovou operaci lokálního výukového bodu a přesunete robota do polohy tohoto bodu.



Obrázek 4.7-61 Spuštění místního výukového bodu

**Podrobnosti:** Kliknutím na ikonu "Podrobnosti" na panelu operací s daty v tabulce místního výukového bodu zobrazíte podrobnosti o místním výukovém bodu.



Obrázek 4.7-62 Podrobnosti o místním výukovém bodu

### 1.3.5.7.16 Aktuální záloha programu

Poté, co uživatel upraví výukový program a klikne na tlačítko uložit, spustí se funkce "zálohování" aktuálního programu (doba zálohování je 1 rok) a původní obsah aktuálního programu se uloží a zobrazí na pravé straně, což je p r o uživatele výhodné pro porovnání upraveného obsahu. Uživatelé mohou zobrazit příslušný obsah zálohy programu výběrem data a kliknutím na ikonu "Odstranit" v pravém horním rohu odstranit obsah zálohy aktuálního programu. Obsah aktuální zálohy programu lze pouze prohlížet, nikoli upravovat.



Obrázek 4.7-63 Aktuální záloha programu

# 1.3.5.7.17 Grafické programování

Vzhledem k tomu, že výukový přívěsek obvykle není připojen k periferním zařízením, jako je klávesnice a myš, může uživatel při přístupu k webové aplikaci robota WebAPP na straně výukového přívěsku upravovat výukový program robota prostřednictvím grafické programovací funkce. Funkční standardizační funkce jsou implementovány pomocí knihovny Blockly, kterou lze integrovat do systému WebAPP, a podle potřeby lze implementovat vlastní bloky kódu a po dokončení programování metodou drag-and-drop je program převeden do programu LUA a vydán a spuštěn prostřednictvím stávajícího instrukčního protokolu.

Díky použití grafického programování může být jednoduchý, snadno pochopitelný, snadno ovladatelný a jazyk může být ovládán v čínštině.

a ≡			Stopped toolcoord0 wobj0 exaxis0 10 🖢 🛦 🔞 🕫	8
😔 initialize <	= 🛯 /	Workspace Name:	Operation&Status 380° Free Mounting Fixed Mounting	
📄 Teaching 🗸	Logic Loop		Joint Base Tool Wobi Move 🔿 Robot Pose	Ы
Program Teaching	Math Variable		Guide 10 TPD FT RCM Joint	its
Graphical Progr	Function		Speed         100         %         J4 1-107 964 JE 1-61.891 JB 139.719           Acceleration         180         */5^2	
Manage Teaching	Control		Threshold 30 * To X :950.303 Y :471.809 Z :546.499	P
-4- Status <	Senior		Single Muß	
명 Auxiliary <				ET.
★ Settings			J2	
			34 107 864	m
			J5 😋 -0 🕕 [91 891] Nett : 0.00.0	
			35	ox.
			Profix 004 005 006 007 Profix 004 005 006 007 C04 005 006 007	
			Foint name 1 Add Dio Dit Di2 Di3	
			Sensor V 000 011 012 013 010 011 012 013 014 015 016 017	

Obrázek 4.7-64 Grafické programovací rozhraní

Stránka je rozdělena do tří oblastí: "operační panel", "panel nástrojů" a "oblast pro úpravu kódu v pracovním prostoru". Celkové rozvržení je následující



Obrázek 4.7-65 Celkové uspořádání grafické programové stránky

**Ovládací** panelTlačítko "Načíst" slouží k načtení pracovní oblasti, funkce tlačítka "Uložit" slouží k uložení bloku kódu jako odpovídajícího výukového programu po úpravě a tlačítko "Vymazat" slouží k rychlému vymazání oblasti pro úpravu kódu;

ToolboxBlok kódu obsahující všechny instrukce a logické kódy lze přetáhnout na pracovní plochu a vytvořit blok kódu a upravit jej; panel nástrojů Toolbox se dále třídí podle typu instrukce. Logické instrukce: if-else, while, print atd.; základní pohybové instrukce: Klasifikace instrukcí podle scénářů použití: lepení, svařování, pásový dopravník atd. Požadovaný blok kódu můžete snadno najít během používání.

Pracovní plochaGrafické bloky kódu lze upravovat a zobrazovat v oblasti pro úpravu kódu.

### 1.3.5.7.17.1 Grafické příkazy pro programování pohybu



Mezi příkazy pro programování pohybové grafiky patří PTPLinARC a další příkazy pro pohyb.

Obrázek 4.7-66 Grafické programování pohybu

# 1.3.5.7.17.2 Ovládání grafických programovacích příkazů

Řídicí grafické programovací příkazy zahrnují WaitIOa další příkazy.

≡ Ā		Stopped toolcoord0 wobj0 exaxis0 10 🖢 🛦 🕅 💅 ⊗
initialize <	🖀 💾 🍠 🛛 Workspace Name:	Operation&Status 360° Free Mounting Fixed Mounting
📄 Teaching 🗸	Logic WaitMs 1000	Joint Base Tool Woby Move .
Program Teaching	Math Variable Mode Manual	Joints J. 19 17 ROM Joints J. 1948 806 12 - 79 438 .05 : 75
Graphical Progr	Function Pouse No function +	Speed         100         1%         J4 :-157 964 JE : 61.891 JB : 99.719           Acceleration         180         75%2
Manage Teaching	Control Weld SetToolList toolcoord0 -	Ttreahold 30 * * * * * * * * * * * * * * * * * *
-뉴 Status < 명문 Auxiliary <	Senior SetEctoolList_etoolcoord0	JI - 138 896 FT
★ Settings	SetWobToolList (wobjcoord0 -	J2
	Set_AO	J3 → -O- + 75 J4 → -O- + 107 864
	Port Ctrl-A00 + , Condition 100 % , Wheather blocked block - ,	J5 😑 -0 🕕 [91 891] Num : 0.00.9
	Whether to apply threads not +	.8
	Get_AI	Prefix C00 C01 C02 C03 C04 C05 C07
	Value 100 % , Wheather blocked block	Point name 1 Add Did Did Did Did Did Did Did Did Did D
	Whether to apply threads not +	Sensor Cuil Con Coz Cola Col4 Cos Cole Col7

Obrázek 4.7-67 Grafické příkazy pro programování ovládání

# 1.3.5.7.17.3 Pokročilé grafické programovací příkazy

Mezi pokročilé grafické programovací příkazy patří příkaz dofilethreadfold a další pokročilé příkazy.



Obrázek 4.7-68 Pokročilé grafické programovací příkazy

# 1.3.5.7.17.4 Příklad použití grafických programovacích příkazů

Po výběru typu grafického programování klikněte na blok grafického kódu, který chcete použít, můžete jej přetáhnout a sešít v pracovním prostoru.

Pokud například vyberete instrukce PTP a Lin motion a řídicí příkazy Waitms pro sešívání, můžete také na vnější vrstvě vnořit pokročilou instrukci skládání a zadat název komentáře, čímž lze realizovat operaci skládání bloku kódu. Kliknutím na rozevírací pole vyberte typ parametru příkazu a do vstupního pole můžete vyplnit údaje parametru příkazu. Příklady grafických programovacích příkazů jsou následující:



Obrázek 4.7-69 Příklad použití grafických programovacích příkazů

Po dokončení spojování grafických programovacích instrukcí a vyplnění parametrů vyplňte název pracovního prostoru a kliknutím na ikonu "Uložit" program uložte. Vyberte zapsaný "Workspace", klepněte na tlačítko Start Run a můžete tento program spustit.

# 1.3.5.7.17.5 Modularizace bloků grafického programového kódu

Pro zlepšení čitelnosti grafických programových kódů je přidána funkce modularizace bloků grafického programového kódu, tj. pokročilé instrukce: skládání bloků instrukčního kódu.

≅ &		Stopped toolcoord0 wobj0 exaxis0	10 6 8 8 2 8
initialize <	🗮 💾 🍠 🛛 Workspace Name:	Operation&Status 360" Free Mounting Fil	ed Mounting
Teaching      Program Teaching     Graphical Progr     Manage Teaching	Logic Loop Math Variable Function Motion Control Fold Weld	Joint Base Tool Wook Move C Eads 10 TPD FT RCM Speed 100 % Acceleration 180 % Threshold 50 *	Robot Pose         Joints           J1 - 118.895.32 : 70.438.41.75         J 300000           J4 : -107.864.35 : 61.891.36 : 302.719         J COP           X : 590.000         Y : 471.809         Z : 566.400           X : 590.000         Y : 471.809         Z : 566.400
-µ– Status < ⊞ Auxiliany < ★ Settings	Senior	Single         Multi           J1         0         ↓ 138 896           J2         -0         ↓ 79 428           J3         -0         ↓ 75	FT Tool Fx:0.000 Fy:0.000 F2:0.000 Act_State:1
		34     →     →     1107 864       35     →     →     →       36     →     →     →       38     →     →     →       36719     →     →	Line-Num Nim : 0.0.0 DD6 DD1 DD2 DD3 D04 DD5 DD5 DD7
		Profix Point name Add Sensor	COA         COA         COA         COA         COA           COA         COA         COA         COA         COA         COA           DiA         DiA         DiA         DiA         DiA         DiA           Di4         Di4         Di5         DiA         DiA         DiA           CIA         CiA         CiA         CiA         CiA         CiA

Obrázek 4.7-70 Blok kódu skládacích instrukcí

• 1. Napište instrukci bloku kódu, přidejte blok kódu skládací instrukce do vnější vrstvy a do vstupního pole napište poznámky k instrukci.

ā ≡		Stopped toolcoord0 wobj0 exaxis0 10 🖢 🛦 🕲 🕫 😣
initialize <	🗮 💷 🍠 Workspace Name: 085145	Operation&Status 360° Free Mounting Fixed Mounting
Teaching      Program Teaching     Graphical Program     Manage Teaching     L Status      Status      Status      Status      Status      Settings	Logic Loop Math Variable Function Control Weld Senior Uveld Senior Uveld Senior Uveld Senior Uveld Senior Uveld Senior Uveld Senior Uveld Senior Uveld Senior Uveld Senior Uveld Senior Uveld Senior S	Joint         Base         Tool         How         Image: Control of the control

Obrázek 4.7-71 Vykreslení skládacích instrukcí

 2. Klikněte pravým tlačítkem myši na "Collapse Block" na panelu operací pravého tlačítka myši, tato část bloku instrukčního kódu se složí, blok kódu se složí do jednoho řádku a zobrazí se a program lze po složení správně spustit.

≡ A		Stopped toolcoord0 wobj0 exaxis0 10 🖢 🛦 🕅 💅 ⊗
😔 initialize <	🚍 🛄 🍠   Workspace Name: 065145	Operation&Status 360° Free Mounting Fixed Mounting
📄 Teaching 🗸	Logic Loop	Joint Bass Tool Wold Move (A)
Program Teaching Graphical Progr	Variable Function	Joints         Joints           Speed         100         %         J1:118.897.12:79.438.15:75           Acceleration         180         79/2         J4:107.964.15:61.85:1.8:39.719
Manage Teaching	Mation Control Weld	Threshold         30         *         TCP           X 1990.301         Y 1471.509         Z 1546.499         X 1903.001         Y 1471.509         Z 1546.499           Smote         Mutr.         RX1-161.306         RY1.12.546         RZ1-96.309
88 Auxiliary <	Senior test123 Told repeat 5 times	JT 😑 🗿 — 🛟 -130.897 🛛 💦 Toa FT
🗙 Settings		J2         →         →         12428         ↓ </th
		J4
		35
		Profix
		Dig         Dig <thdig< th=""> <thdig< th=""> <thdig< th=""></thdig<></thdig<></thdig<>

Obrázek 4.7-72 Obrázek efektu po složení

3. Přejetím myši realizujte funkci zvětšení stránky, konkrétní efekt je následující:

≅ &		4							Stopped	toolco	ord0	wobj0	exaxis0	10	6	A	8	12	8
initialize <	= B Ø	Workspace	Name: ces145			a	- 0.	w 20 x		- 10 - W			NC ELSE	a - 55					~
📄 Teaching 🗸	Logic Loop		* *** * **		x 0.8									0.18		18	15		с. ж
Program Teaching	Math Variable						test	inpeur S	tittes				2.8						
Graphical Progr	Function							testyd	15	Smooth transit	ion + D	. Latel . C	ot 🗊						8.8
Manage Teaching	Control	Te	est123			w .:		PTP C	ntyyh20 - 11	10 Stop	0 not 1	1							
_}_ Status <	Senior		DULL D	Income						-2.11.0		24 1 1 1							
88 Auxiliary <		5 A 8	old repe	at 5	times														
🗙 Settings			do	LIN		nint et									10				
		8 B.		testvvh	15 . 1	00	Smo	oth tr	ansitic	n -	0	not		not					
		20.12		(cost) jii		, 100	Canto	ounte	uniontic			mer		101	-				
		8 S 👘		WaitMs	1000	S. 8	12	1.2.2	2 2 3	波 武	8.8	10							
		8 A.		PTP te	styyh20	D . C	100 ,	Stop	• , C	), no	ot 🔹	1							8.8
		0 C 📕		PTP te	stvvh23		100	Stop		) no	t -		8 (19) ( 1						
				5 8 1	8 S 8	22 8		5 8 s	8 <b>35 5</b> 3		85 12	6 61	e testi						
				1.1.1										2 5					

Obrázek 4.7-73 Schéma efektu funkce zvětšení složené stránky

# 1.3.5.7.18 Řízení výuky

Kliknutím na "Správa výuky" zobrazíte všechny uložené informace o výukových bodech. V tomto rozhraní můžete importovat a exportovat soubory výukových bodů. Po výběru výukového bodu kliknutím na tlačítko "Delete" (Odstranit) informace o bodu odstraníte. Hodnoty výukových bodů x, y, z, rx, ry, rz a v lze upravit. Zadejte upravenou hodnotu, zaškrtněte modré políčko vlevo a kliknutím na horní modifikaci upravte informace o výukovém bodu. Kromě toho mohou uživatelé vyhledávat výukové body podle názvu.

ā ≡			(	<del>9</del> 10									Stopped	toolcoord0	wobj0	exaxis0 10	-		8	10
🕏 Initialize 🖌	Teachi	ng Manage	ement																	
🖹 Teaching 🗸	Import	Export	Modity	Dekte																
Transmission inter-	Name Se	search																		
Program teaching	Ω	Name	x	۲	z	RX	RY	RZ	J1	JŽ	J.	.,4	.35	.16	TOOL	WOBJ		٧	Opr	erate
Graphical Program	D	Ttart .	-547.837	29,059	123.973	-91,794	0.770	-92.430	-28.369	-72,041	121.795	-50.526	62,839	-44,613	tootcoord	3 workprecer		100	۲	• •
Manage Teaching	Ð	10411	-547.844	30.405	117 171	-76.793	0.771	-91.429	-28.611	-74.362	105.261	-14.768	62,808	-52,196	toolcoord	3 workpiecel	8	100	۲	
4⊢ Status	D	Ita/12	-517.841	38.408	.111.161	-70.794	0.771	-91.429	-30.962	-78.242	105.385	-3.892	61.545	-56.122	toolcoord	workpiecel		100	۲	• •
88 Aunitary <	12	15813	-480.315	30.410	111.159	-70.794	0.772	-91.429	-35.660	-85.664	114.451	-4.140	\$6.328	-58.626	toolcoord	a workpiecel	5	100	۲	
🗙 Settings	D	1tail:	-455.679952	39 326639	113.092445	-81.678628	1.060899	-90 377373	-49,159	-91,011	133.024	-27.570	41.072	-56.042	toolcourd	3 workpieces	5	100	۲	•
	D	twort	98 956897	-312 020508	593.540405	-179 678940	-1.963225	0.1897.47	89 638	-92.426	119 538	-115.693	90,734	-43.580	tookoord	3 workpiecer	5	100	۲	
	D	1wu2	96.795090	-308.837006	593,592957	178.072754	-2.575468	93.845573	89.638	-92,426	119.538	-115.694	90.734	50.009	toorcores	3 workpiecer	£	100	۲	
	D	2tai1	-546 226	01.892	287.717	-90.354	0.549	-92.634	-28 909	-89.149	95.347	-5.354	62.520	-45.103	toolcoord	3 workpiecel	2	100	۲	•
	D	26412	-541 231	32 385	285 692	-74.494	0.550	-92.633	-28.182	-80.515	68.852	50 115	64.541	-52.563	toolcoord	workpiecel		100	æ	•
	ъ.	71413	-511 225	12.367	265 805	-69.452	0.552	-92 634	-30,797	-53.359	71.943	15.574	62.987	-55.964	toolcoord	a workpiecel		100	æ	()
		25414	-461 290	32 385	265 810	.69.456	0.554	.92.634	-39-117	-54 253	83,490	37.394	35,475	-60 133	tooicoord	a workbiecer		100	æ	
	-	2aut	117 120578	.374.456.517	635 139648	179 304011	-1.420023	5 3792 19	02.802	.101 524	117.062	-105 141	90.201	.43.501	Independent			100		0
	0	David	62 944570	-303 760946	768 197758	.05 790657	.9 (123304	173.015533	82.528	.05 199	76.007	.71 208	50.024	49.597	tookoocer	waterioral		100		ŏ
	10	Amos.	750 710510				-2.020004	88.75.775	00.000	-90,100	70.457	-117.240	24.205		Independent			100		
	10	Thread	200.740019	-120.100702			0.830000	00.001218	00.120	-00.100	474.005	400.000	45.044	40.014	Instances	. mungrettet		100		0
Successful	199	2404	202.280099	109.007091	401.703479	05.299/03	0.030905	-89.201702	2.929	-194.894	124.066	-108.822	eb.501	43.662	Wareborg.	s montprecer		100		0
<ul> <li>Acquisition of learning point</li> </ul>		200	252,215651	-108.012054	620.447693	89.274215	0.053055	-89.201569	2.930	-123.551	87/048	-143.143	\$6,500	-43.640	toolcoord	a workpiecel	2	100	۲	0
( Cittar	0	2WU6	270.723328	-107.069771	516.709460	69.271538	-0.210132	-89.263046	2.930	-123 662	87.648	-163.159	86.649	-43.640	toolcoord	a workpiecel	3	100	۲	0

# Obrázek 4.7-74 Rozhraní pro správu výuky

Podrobnosti: Kliknutím na tlačítko "Podrobnosti" zobrazíte podrobnosti o výukovém bodu.

a ≡			-	4								3	Stopped 3	Obracation	wab)0 ex	antaŭ 10			8
O Inteller C	Teachin	g Manage	ment					details		×									
🖹 Teaching 🔍	( Import	Expect	Modify	-2000			1	X -547 837											
Prophin Seatting	name des	eti						Y: 29.059 Z: 123.973											
Coolical Pausian	191	Name	1	· *	1	<b>派</b>		RX: -91.794 RY: 0.770			30		3	*	TOOL	VUBJ	1.4	Operate	
Weinstein ausgehörten.	( <b>D</b> )	10000	-617.637	28.009		00,204		RZ: -92.430		MIC	321398		\$2,359	44.012	100004403	BURGEOR	100		
Manage Tasching	10	-1011	017 814	30 400		75.792		J2 -72 041		102	105 251	-14.758	62,905	32.196	-	anterest.	100		
A-Bina K	10	: Ret12	317.841	38.408	111.161	70.794		J4: -50:526		242;	101 305	4.000	45.540	-56-123	Inchorem)	workpieceG	100		
III Southery €	12	1000	-400.015	20.410	-	20.794		J5: -44.613		640	314 401	40440	(64)(62)	- 26 204	tockocetti	antipecro	(100		
🛪 Settrije	0	1100			111(05445	61.875538	3	WOBJ: workpiecel	0	an.	11104		41.072	51.045	Transform	minipresso.	100		
	D.	11615	98.806897	1512 (00)528	001 540405	675 070940		ACC: 180		426	110 558	093	(0754	45.500	INNODECO	wittightor	(100)		
	101	(test)	Se Visitio	-308 032009	2943 5927057	178.823754		E2: 0.000		120	199356	-05494	MI734	30.009	lineosees.	autorero	(200)		
		- 2144	-640.225	24.002	207.717	90.514		E4 0.000		743	30.347	4.994	42.530	48,903	luskor	wastplaceO	(102)		
	121	2102	-041231	10.385	288, 692	24264					101102	58-115	64.541	32515	Indenteiß	entrépieceő	1400		
	12	7005	-011.775	39-3927	265.305	472 413				104	75.543	20.774	12.167	32.964	tinocopents	windowero .	107		
	:D)	- 2004 (	-411.220	02.300	255.810		() 554	-05.504	-381 M P (	-54.200	100.400	47.004	38.47k	-46.133	10060100	workpiece0	:100		
	18	Sect	117 130874	-378 (2001)7	155 199648	179.500031	-1 (2001)3	5.57016	30.005	-107 532	112.002	-initian	50.201	-12 502	liniorets	(inighter)	100		
	10	244	62 344970	-000 2002018	766 TH 7798	-05.725807	-21523300		NU. 5221	-16. TH	78.097	-71.228	90.028	41.507	TOSKOLIOT	MUNDROLD			
	(D)	2000	200,745018	-228.066712		18311194	-2.070543	Constantine -	59 th	-95,139	10.097	102146	19295	41814	<b>Weikeerst</b>	etrigenete	1900		
	0	See.	257.285498	101.007091	467 25:3479	-	-	-	1577	154 254	124.000	164 822	16 601		tismoort?	-	100		
	100	2415	258,210801	1001012014	120 447693	88,274246	0.053606	89,201069	2,300		87.548	41143	16.655	-49.640)	lockote(D	auspicce0	100		
	15	. Janut	ano.taboas	-107 SHATTS		88.271036		-00.253040	2.900		87.048	-100.105	05.5-03	43.640	INCACOUNTS.	- monuteried	100		

Obrázek 4.7-75 Detaily bodu učení

**Spusťte**: Klepnutím na tlačítko "Start Run" provedete jednobodovou operaci místního výukového bodu a přesunete robota do polohy tohoto bodu.

las z	Teachin	g Manager	ment															
hing V	Import	Export	Modity	Delete														
	Name Ben	inch.																
n Teaching	0	Name	x	Y	z	RX	RY	RZ	J1	.J2	JJ.	.14	.35	./6	TOOL	WOBJ	v	Opera
l Piogram	D	Itari	-547.837	29.009	123.973	-91.794	0.770	-92.430	-28.369	-72.041	121.796	-50.526	62.839	-44.613	tookoord3	workpieceo	100	
Teaching	D	16411	-547.844	30.405	117 171	-75.793	0.771	-91.429	-28.511	-74.962	105.261	-14.768	62,808	-52 196	toolcoord3	workpiece0.	100	۲
<	D	11a/12	-517.841	38.408	.111.161	-70.794	0.771	-91,429	-30.962	-78.242	105 385	-3.892	61.545	-56,122	Eprocologi	workpiece0	100	۲
ny ≮		10013	-480.315	20.410	111.159	-70.794	0.772	-91.429	-35.660	-85.664	114.451	-4.140	56.329	-58.826	tooicoord3	wompiece0	100	۲
16	D	1tail:	-455.679352	39 326839	113.092445	-81.676528	1.060833	-90.377373	-49,155	-91,011	133.024	-27.570	41.072	-56.042	toolcourd3	workpiece0	100	۲
		twut	98 956897	-312 020508	593.540405	-179 678940	-1.963225	0.1897.47	89.638	-92.426	119 538	-115.693	10.734	-43.580	tookcoord3	workpiece0	100	۲
		1wi/2	96.795090	-308.837006	593,592957	178.072754	-2.575468	93.845573	89.638	-92,426	119.538	-115.694	90.734	50.009	toolcoord3	workpiece0	100	۲
	D	Start	-546 226	01.092	287.717	-90.354	0.549	-92,634	-28,909	-09.149	95.347	6.354	62.520	-45.103	toolcoord3	workpiece0	100	۲
	D	2fat2	-541231	32.385	285 692	-74.494	0.550	-92.633	-28.182	-80.515	68.852	50.115	64.541	-52.563	toolcoord9	workpiece0	100	۲
	12	2ta/3	-511 225	32.387	265.805	-69.452	0.552	-92.634	-30.797	-63.359	71.943	35.874	62.987	-55.964	toolcoord3	workpiece0	100	۲
		21414	-461.290	22.386	266.810	-69.466	0.554	-92.634	-39.117	-54,233	83,490	37.334	66.476	-60.133	toolcoorda	workpiece0	100	۲
		2wu1	117 120674	-274.439117	635.139648	179.304033	-1.420023	3 3792 19	92.882	-101 524	117.062	-105 141	90.201	-43.593	Essociation (	workpiece0	100	۲
	Q	2wu2	62,344570	-300.760986	766.137756	-93,729637	-2.023304	173.015533	82.528	-95.139	76.097	-71.226	90.024	-43.587	toolcoordt	workpiece0	100	۲
	D	2wu3	259.740819	-228.166702	593.047913	-0.311194	-2.073543	65.351279	69.120	-90.139	75.097	-157,140	21,295	-45.614	toolcoord1	workpiece0	100	۲
		2wµ4	252,285599	-108.507591	487.753479	B9.269753	0.530905	-89 201752	2.929	-134.894	124.066	-168.822	86.661	43.662	toolcoord3	workpiece0	100	۲
steps of		2845	252,215851	-108.512054	620.447693	89.274215	0.553055	-89,201569	2.930	-123.551	87/048	-143.143	86,550	-43.640	tooicoord3	workpiece0	100	۲
and been	0	2wu6	270.723328	-107.069771	516.705460	69.271538	-0.210132	-89.263046	2.930	-123 562	87.648	-163.159	86.649	-43.640	toolcoord3	workpiece0	100	۲

Obrázek 4.7-76 Spuštění bodu učení

Důležité: Upravené hodnoty výukových bodů x, y, z, rx, ry, rz by neměly překročit pracovní rozsah robota.

#### 1.3.5.8 Informace o stavu

#### 1.3.5.8.1 Systémový protokol

Kliknutím na tlačítko "System Log" (Systémový protokol) v části "Status Information" (Informace o stavu) na levém panelu nabídky vstoupíte do rozhraní pro zobrazení protokolu. V protokolu jsou zaznamenány některé důležité záznamy o provozu výukového přívěsku, jako je přihlášená osoba, zvýšení a snížení výukového bodu atd. Po kliknutí na vstup se standardně zobrazí záznamy protokolu aktuálního dne. Pokud se chcete dotázat na předchozí záznamy, vyberte cíl v poli "Výběr data" Datum, níže se zobrazí záznam protokolu aktuálního dne v reálném čase a počet záznamů protokolu lze nastavit v nastavení systému pro podrobnosti. Pokud je informací o uživatelském záznamu příliš mnoho, může uživatel Vyhledat příslušné informace o záznamu podle typu.

a =	(	* •		Stopped toolcoord0 wobj0 exaxis0 0 🖢 🛦 🕅 😂
🕸 Initialize 🗸	System Log			
🔲 Teaching ⊀	Date selection 2023-	03-22	Export	
⊿⊶ Status ∨	whole          error          on         i         error         i         i         error         i         i         i	varning 🔘 General open	ation 💿 Apply operati	ion   Robot operation
System Log	time	classification	operator	content
Status Query	09:25:02	General operation	admin	User login succeeded
	06:45:50	General operation	admin	User login succeeded
昍 Auxiliary く	06:10:34	General operation	admin	User login succeeded
X Settings				

Obrázek 4.8-1 Rozhraní systémového protokolu

# 1.3.5.8.2 Stavový dotaz

Kliknutím na nabídku "Status Query" (Dotaz na stav) v levém panelu nabídek "Status Information" (Informace o stavu) vstoupíte do rozhraní pro dotazování na stav, jak je znázorněno na obrázku 4.8-2 Status Query.

ā ≡	×   •	Stopped toolcoord0 w	robj0 exaxis0 0 🖢 🗛 🕲 🐲 🛞
🐵 Initialize \prec		Parameter Configuration	Start Querying the Ribbon Q Query
🗐 Teaching 🖌	Joint Base Tool Wobj Move	Ribbon	
-∕⊧- Status →	Eaxis IO TPD FT RCM		No query
System Log	tpd track record		
Cystem Log	Track na		
Status Query	Posture Joint posture *		
	Period 2 T		
믬 Auxiliary <	Di config nothing •		
V Collinse	Do config nothing *		
A Settings	TPD-State Not recording		
	Config. Start Stop		
	Tpd track editing		
	Track na V Get points		
	Start		
	o		
	End		
	o		
	Simulate		
	Track na Delete		

Obrázek 4.8-2 Stavový dotaz

Kroky operace dotazu na stav:

Krok1Kliknutím na tlačítko "Figure" (Obrázek) zobrazte vyskakovací okno nastavení obrázku, jak je znázorněno na obrázku3.8-3Nastavení obrázku, vyberte možnost Dotaz na zobrazení obrázku pro typ dotazu, vyberte parametry, které mají být dotazovány, a obrázek, ve kterém jsou parametry umístěny v nastavení obrázku, a klikněte na tlačítko "Přesunout doprava" Parametry lze konfigurovat v obrázku. Kliknutím na tlačítko "Settings" (Nastavení) vydáte příkaz Figure setting (Nastavení obrázku). V současné době může pouze jedna tabulka obsahovat maximálně čtyři parametry, které mají být zjišťovány, a lze nastavit maximálně jeden Figure;

Please select a query		
data frame Joint-pos command Joint-pos feedback	<b>→</b>	
End-pos command Tool-pos command End-pos feedback	×	
Tool-nos feedback		

Obrázek 4.8-3 Nastavení obrázku

Krok2Funkci spouštěče není třeba dočasně nastavovat a na data se lze dotazovat kliknutím na tlačítko "Dotaz".

### 1.3.5.9 Pomocná aplikace

#### 1.3.5.9.1 Upgrade systému

Na liště nabídky "Robot Body" v "Auxiliary Application" klikněte na tlačítko "System Upgrade", čímž vstoupíte do rozhraní pro aktualizaci systému. Upgrade systému se dělí na upgrade softwaru, upgrade ovladačů a vypnutí systému.

**Upgrade softwaruKlikněte na** "Upload File" v části Software Upgrade, vyberte aktualizační balíček software.tar.gz na disku U, klikněte na Upload Upgrade Package a "Uploading. . . Upload Percentage" se zobrazí vedle tlačítka upgradu. Po dokončení stahování souboru na pozadí se v rozhraní zobrazí zpráva "uploading completed, upgrade in

progress", zkontrolujte číslo MD5 a číslo verze souboru, po průchodu dešifrujte a dekomprimujte soubor upgradu a zobrazí se výzva "Upgrade je úspěšný, restartujte prosím ovládací panel!", pokud dojde k detekci , rozbalení nebo jiným chybám, a vedle tlačítka upgradu se zobrazí "upgrade se nezdařil".

Software upgr	ade	
Choose File	No file chosen	
		Upload

Obrázek 4.9-2 Upgrade systému

**Důležité:** Název balíčku pro aktualizaci softwaru je určen jako software.tar.gz. Pokud je název aktualizačního balíčku v rozporu s ním, aktualizace se nezdaří. Stačí jej změnit na určený název aktualizačního balíčku.

**Aktualizace firmwaruPo** vstupu robota do režimu BOOT nahrajte komprimovaný balíček aktualizace, vyberte podřízené jednotky, které je třeba aktualizovat (podřízené jednotky řídicí jednotky, podřízené jednotky pohonu hlavního tělesa 1~6 a koncové podřízené jednotky), proveďte operaci aktualizace a zobrazte stav aktualizace.

Firmware update	
	Enter boot
Upgrade control box	
Upgrade Joint 1	
Upgrade Joint 2	
Upgrade Joint 3	
Upgrade Joint 4	
Upgrade Joint 5	
Upgrade Joint 6	
Upgrade End	
Choose File No file chosen	
	Upload

Obrázek 4.9-3 Aktualizace firmwaru

Aktualizace **konfiguračního souboru podřízených zařízeníPo** vypnutí robota nahrajte soubor aktualizace, vyberte podřízená zařízení, která je třeba aktualizovat (podřízená zařízení řídicí jednotky, podřízená zařízení pohonu hlavního tělesa 1~6 a koncová podřízená zařízení), proveďte operaci aktualizace a zobrazte stav aktualizace.



Obrázek 4.9-4 Aktualizace konfiguračního souboru Slave

### 1.3.5.9.2 Zálohování dat

Na panelu nabídek "Robot Body" v "Auxiliary Application" klikněte na "Data Backup", čímž vstoupíte do rozhraní pro zálohování dat, jak ukazuje obrázek 3.9-5.

Data záložního balíčku obsahují data souřadnicového systému nástroje, konfigurační soubory systému, data výukových bodů, uživatelské programy, šablonové programy a uživatelské konfigurační soubory. Když uživatel potřebuje přenést příslušná data tohoto robota do jiného robota, může použít tuto Funkce je realizována rychle.

data.
pload

Obrázek 4.9-5 Rozhraní pro zálohování dat

### 1.3.5.9.3 10s datový záznam

Na panelu nabídky "Robot Body" v "Auxiliary Application" klikněte na "10s Data Recording", čímž vstoupíte do rozhraní funkce 10s záznamu dat.

Nejprve vyberte typ záznamu, který se dělí na záznam výchozích parametrů a záznam volitelných parametrů. Záznam výchozího parametru jsou údaje automaticky nastavené a zaznamenané systémem a záznam volitelného parametru si uživatel může zvolit údaje parametru, které se mají zaznamenat. Maximální počet parametrů je 15. Po výběru seznamu parametrů vyberte parametr záznamu a kliknutím na tlačítko "Přesunout doprava" nakonfigurujte parametr do seznamu parametrů. Kliknutím na tlačítko "Start Recording" (Spustit záznam) zahájíte záznam dat, kliknutím na tlačítko "Stop Recording" (Zastavit záznam) zastavíte záznam dat a kliknutím na tlačítko "Download Data" (Stáhnout data) stáhnete data za posledních 10 sekund.

lease select a r	ecord •
data frame	•
Joint-pos	lă l
command	
Joint-pos	×
eedback	
End-pos	
command	
Tool-nos	

Obrázek 4.9-6 10s datový záznam

### 1.3.5.9.4 Konfigurace výukového bodu

V nabídkovém panelu "Robot Body" v "Auxiliary Application" klikněte na "Teaching Point Configuration" a vstupte do rozhraní pro konfiguraci výukového bodu.

Před použitím tlačítkového pole nebo jiných IO signálů pro záznam funkce výukového bodu uživatel nejprve nakonfiguruje předponu názvu výukového bodu, horní hranici počtu a metodu výuky. Předpona názvu podporuje dva režimy: vlastní předpona a aktuální název programu jako předpona. Například přizpůsobte předponu názvu "P", horní limit počtu "3", metodu výuky "výuka robota", zaznamenávejte postupně aktuální koncové (nástrojové) body robota: Při dalším záznamu se přepíší předchozí body záznamu: P1, P2, P3.

Teaching p	oint configuration	
Teaching po	int configuration	
Name pr	Custom prefix •	
Custom	test	
Number	10	
Teachin	Robot teachin •	
		Set

Obrázek 4.9-7 Konfigurace bodu učení

# 1.3.5.9.5 Maticový tah

V nabídce "Robot Body" v "Auxiliary Application" klikněte na "Matrix Movement" a vstupte do rozhraní pro konfiguraci funkce maticového pohybu.

Tato funkce řídí pravidelný pohyb manipulátoru nastavením tříbodových souřadnic a hodnot řádkové a sloupcové vrstvy a výšky vrstvy, což je vhodné pro běžné paletizační aplikace. Prvním krokem je výběr režimu pohybu robota, "PTP" nebo "Line", druhým krokem je nastavení dráhy pohybu robota, "head- to-tail walking method" nebo "bow walking method", třetím krokem je nastavení metody stohování, "stacking stacking" nebo "unstacking".

Manipulator	motion mod	le	
Motion s	PTP	×	
Robot motic	on path		
Path sel	Head to ta	il w 🔻	
	0	Ŷ	
	25 25	P	
Stacking mo	ode setting		
	stacking	*	

#### Obrázek 4.9-8 Přesun matice

Čtvrtým krokem je výuka tří bodů podle cesty. První bod je výchozím bodem první řady a podle tohoto bodu se určuje držení paže v průběhu celého pohybu. Druhý bod je koncovým bodem první řady a třetí bod je koncovým bodem poslední řady. Pátým krokem je nastavení počtu oMěřítek a sloupců. Šestým krokem je nastavení počtu vrstev a výšky každé vrstvy. Posledním krokem je pojmenování souboru programu pohybu matic a program pohybu matic je úspěšně vygenerován.

First point:	cvrCatchPoint •	Apply
Second	cvrCatchPoint •	Apply
Third poi	cvrRaisePoint •	Apply
Number of I	ayers and height	
Number of la	ayers and height	mm

Set three points according to the path

Obrázek 4.9-9 Přesun matice

### 1.3.5.9.6 Původ práce

V nabídkovém panelu "Robot Body" v "Auxiliary Application" klikněte na "Working Origin", čímž vstoupíte do rozhraní funkce konfigurace pracovního původu.

Na této stránce se zobrazí název a informace o společné poloze původu práce. Původ práce je pojmenován pHome. Kliknutím na tlačítko "Nastavit" použijete jako pracovní počátek aktuální pozici robota. Kliknutím na "Move to this point" (Přesunout do tohoto bodu) přesunete robota do pracovního počátku. Kromě toho je v konfiguraci DI přidána konfigurovatelná volba přesunu k původu práce a v konfiguraci DO je přidána konfigurovatelná volba oDosažení původu práce.
Job	Origin C	onfigu	iration		
Job	Origin				
Poin	it Na j	oHome			
J1	NaN	J2	NaN	J3	NaN



### 1.3.5.9.7 Konfigurace rušivé zóny

Na panelu nabídek "Robot Body" v "Auxiliary Application" klikněte na "Interference Area Configuration", čímž vstoupíte do rozhraní funkce konfigurace rušivé oblasti.

Nejprve je třeba nakonfigurovat režim rušení a operaci vstupu do rušené oblasti. Režim interference se dělí na "interferenci v ose" a "interferenci v krychli". Pokud je aktivován, zobrazí se aktivační znak. Nejprve zadejte konfiguraci pohybu v interferenční zóně "pokračovat v pohybu" nebo "zastavit".

Interfere Shaft interfere • not close Enter the interference zone motio configuration	t active oper
close Enter the interference zone motio configuration	oper
Enter the interference zone motio configuration	
	n
Sports S Keep moving 🔻	

Obrázek 4.9-11 Konfigurace rušivé zóny

Dále nastavte konfiguraci přetahování do oblasti rušení. Uživatelé mohou nastavit strategii po vstupu do interferenční oblasti v režimu přetahování podle svých potřeb, aniž by omezovali přetahování, zpětné vyvolání impedance a přepnutí zpět do ručního režimu.

rag	j Str	Impe	dance ca	•	
					Set
				26	
0.04	anco	callbac	parame	toreott	ing
npe	edance	callbac	k parame	ter sett	ing
npe J1	edance NaN	J2	k parame NaN	eter sett	ing NaN

Obrázek 4.9-12 Konfigurace přetahování rušivé oblasti

Chcete-li zvolit rušení osy, musíte nakonfigurovat parametry rušení osy. Metoda detekce se dělí na dva typy: "příkazová poloha" a "zpětnovazební poloha". Režim oblasti rušení se dělí na dva typy: "rušení v rozsahu" a "rušení mimo rozsah". Dále Nastavte rozsah každého kloubu a zda je povolen každý rozsah kloubu, můžete zadat hodnotu nebo můžete zaznamenat aktuální polohu robota prostřednictvím tlačítka "Robot Teaching" a nakonec klikněte na tlačítko Apply.

	Teaching	Teaching	
J6	NaN	NaN	Not en: •
J5	NaN	NaN NaN NaN	Not en:  Not en:
<b>J</b> 4	NaN		
J3	NaN		
J2	NaN	NaN	Not en: •
<b>J1</b>	NaN	NaN	Not en: •
	min	max	enable
Interfere	Interfere	ence w 🔻	
Test met Comma		and po: •	
		2	

#### Obrázek 4.9-13 Konfigurace rušení osy

Chcete-li zvolit rušení krychle, musíte nakonfigurovat parametry rušení krychle. Metoda detekce se dělí na dva typy: "pozice příkazu" a "pozice zpětné vazby". Režim oblasti rušení se dělí na "rušení v dosahu" a "rušení mimo dosah". Systém se dělí na "základní souřadnice" a "souřadnice obrobku", které lze vybrat a nastavit podle skutečného použití. Dále nastavte nastavení rozsahu. Nastavení rozsahu se dělí na dva způsoby. Nejprve se podívejte na první metodu "dvoubodovou metodu", která se skládá ze dvou diagonálních vrcholů krychle. Polohu můžeme zaznamenat prostřednictvím vstupu nebo učení robota. Nakonec klepněte na tlačítko Použít.

leachin	Two-j	point met	•	
Enter the m	in <mark>i</mark> mum	value po	oint	
	P		~	7
Maximum	Ŕ		X axis	Zaxis
	$\leq$	Y axis		Minimum
				-
X NaN	Y	NaN	Z	NaN
				Teachin
Enter the m	aximum	n value po	oint	
X NaN	Y	NaN	Z	NaN
				Teachin
			100	

Obrázek 4.9-14 Interferometrická konfigurace krychle

Dále se podívejte na druhou metodu "středový bod + délka strany", to znamená, že středový bod krychle a délka strany krychle tvoří interferenční oblast a my můžeme zaznamenat polohu prostřednictvím vstupu nebo učení robota. Nakonec klepněte na tlačítko Použít.

Range sett	ing		
Teachin	Center point+:	•	
Center poir	nt		
	Y-	X	Zaxis
	z Centerpoint	/ X axis	
l.			$\searrow$
	Y axis		
V	Veres	7 7	iere 1
X NaN	Y NaN	_ Z	NaN
			Teaching
Side length	n of each axis		
X NaN	Y NaN	Z	NaN

Obrázek 4.9-15 Interferometrická konfigurace krychle

## 1.3.5.9.8 Konfigurace svorek LED

V nabídkovém panelu "Robot Body" v "Auxiliary Application" klikněte na "End LED Configuration", čímž vstoupíte do rozhraní funkce konfigurace barvy koncové LED.

Nastavitelné barvy LED jsou zelená, modrá a bílá azurová. Uživatelé si mohou podle svých potřeb nakonfigurovat barvy LED v automatickém režimu, ručním režimu a režimu přetahování. Různé režimy nemohou mít nakonfigurovanou stejnou barvu.

Mode Color	Configuration	rmina	i lea
		·	
Automati	blue	۲	
Manual	green	۲	
Dra <mark>g Mo</mark>	White cyan	۲	
			Configure

Obrázek 4.9-16 Konfigurace LED svorek

### 1.3.5.9.9 Periferní protokol

V nabídkovém panelu "Robot Body" v "Auxiliary Application" klikněte na "Peripheral Protocol", čímž vstoupíte do funkčního rozhraní konfigurace periferního protokolu.

Tato stránka je konfigurační stránkou pro protokol periferie a uživatel na ní může konfigurovat protokol podle aktuálně používané periferie.

agreement	
Protocol Configuration	
ieral protocol is external axi	5
Extension axi: •	
	Set
	agreement Protocol Configuration leral protocol is external axis Extension axis T

Obrázek 4.9-17 Konfigurace periferního protokolu

Přidejte rozhraní lua pro čtení a zápis registrů na základě komunikace Modbus-rtu do výuky programu, adresa vstupního registru 0x1000, počet oMegisters je 50, celkem 100 bajtů datového obsahu; držte adresu registru 0x2000, počet oMegisters je 50, celkem 100 bajtů datového obsahu.

ModbusRegRead(fun\_code, reg\_add, reg\_num): čtení registru;

fun\_code: kód funkce, 0x03-držící registr, 0x04-vstupní registr reg\_add:

adresa registru

reg\_num: počet oMegisters

ModbusRegWrite(fun\_code, reg\_add, reg\_num, reg\_value): zápis registru;

fun\_code kód funkce, 0x06-jediný registr, 0x10-více registrů reg\_add:

adresa registru

reg\_num: počet oMegisters

reg\_value: pole bajtů

ModbusRegGetData (reg\_num): Získat data registru;

reg\_num: po $\check{c}\text{et}$ o<br/>Megisters

Popis návratové hodnoty:

reg\_value: proměnná pole

Ukázka programu:

<b>F</b> ₹ ≡	Stopped tool1 wobj1 exaxis1	100 0 🗚	8 5 8
Initial Setup <	📹 🗟 土 土 🖺 😫 🌾 🖲 亩 스 🗢 邟 Joint Base Tool Wobj Move 🚮	Robot Pose	ah
🗐 Teach Simul~	1 fun_code1 - 8x18 - Eaxls IO TPD FT RCM	10	Joints
Program Teach	Yes         2         fun_code2 - 8x84         Speed:         100         %           PTP         LIN         3         addr - 8x1000         acr         180         1672	J1:0 J2:0 J4:0 J5:0	J3 : 0 J6 : 0
Graphical Pro Teach Manage	ARC         Circle         5         ret = ()         S.Move         M-Move	X : 0 Y : 0 RX: 0 RY: 0	TCP Z : 0 RZ: 0
J⊶ Status Info ≺	Spiral N-Spiral 7 WaltMs(18)	Tool	FT
System Setting	Spline N-Spline 9 WaitMs(18)	Fx:0 Fy:0 Tx:0 Ty:0 Act_State:	Fz:0 Tz:0
	IN         IN<	Num : 0	Line-Num
	TPD     ToolList     13     rets1 - ret[1]       Image: Transmission of the second	000 001	CtriBox
	Mode         Var         15         rets3 = ret[3]         Prefix           CO         ↑         16         rets4 = ret[4]         Prefix	C00 C01 C04 C05	CO2 CO3 CO6 CO7
	While IT_Else 17 end +	DI0 DI1 DI4 DI5	DI2 DI3 DI6 DI7

Obrázek 4.9-18 Příklad programu lua pro komunikaci Modbus-rtu

## 1.3.5.9.10 Konfigurace hlavního programu

Na liště nabídky "Robot Body" v "Auxiliary Application" klikněte na "Main Program Configuration", čímž vstoupíte do rozhraní funkce konfigurace hlavního programu.

Konfiguraci hlavního programu lze použít ve spojení s konfigurací DI při spuštění hlavního programu. Nakonfigurovaný hlavní program je třeba nejprve zkušebně spustit, aby byla zajištěna bezpečnost. Po konfiguraci příslušného DI v nastavení robota pro spuštění funkce signálu hlavního programu může uživatel ovládat signál DI pro spuštění hlavního programu.

main program	
There is currently no default configurator	
Whether yes	
Select m left_pattern_a •	
total_start_time =	*
GetSysVarValue(s_var_8)	
layer_index = GetSysVarValue(s_var_4)	
box_index = 0	
SetSysVarValue(s_var_5, box_index)	* 1/
*Please test run the main program in program	
teaching before configuring it	

Obrázek 4.9-19 Hlavní konfigurace programu

## 1.3.5.9.11 Zámek přetažení

Na panelu nabídky "Robot Body" v "Auxiliary Application" klikněte na "Drag Lock", čímž vstoupíte do rozhraní pro konfiguraci funkce drag teaching lock.

Pro výuku vlečení se přidává funkce blokování stupňů volnosti. Pokud je přepínač funkce výuky tažením nastaven do povoleného stavu, parametry jednotlivých stupňů volnosti se projeví, když uživatel robota táhne. Například když je parametr nastaven na X:10, Y:0, Z:10, RX:10, RY:10, RZ:10, přetahování robota v režimu přetahování může omezit pohyb robota pouze ve směru Y, v případě potřeby Zachovat polohu robota při přetahování beze změny a pohybovat se pouze ve směrech X, Y a Z. Hodnoty X, Y a Z můžete nastavit na 0 a hodnoty RX, RY a RZ na 10.

Drag	g teachi	ng loci	k config	uration	1
Drag	g lock				
Para	ameter s	etting o	of each c	legree	of freedom
Х	NaN	Y	NaN	Z	NaN
RX	NaN	RY	NaN	RZ	NaN
Drag	g the tea	ching fi	unction	De e	Apply enable
				-	Apply

Obrázek 4.9-20 Konfigurace zámku Drag teach

### 1.3.5.9.12 Odborná knihovna pro svařování

Kliknutím na panel nabídky "Welding Expert Library" v "Auxiliary Application" (Pomocné aplikace) vstupte do funkčního rozhraní knihovny odborníků na svařování. Knihovna odborníků na svařování je rozdělena do čtyř částí: tvar svařence, konstrukce dílu, struktura přípravku a konfigurace.

Kliknutím na "Direct Welding" (Přímé svařování) v části "Weldment Shape" (Tvar svařence) vstoupíte do rozhraní pro přímé navádění svařování. Na základě konfigurace základních nastavení robota můžeme pomocí několika jednoduchých kroků rychle vygenerovat výukový program svařování. Zahrnuje především následujících pět kroků. Vzhledem ke vzájemnému vyloučení funkcí je skutečných kroků pro vygenerování výukového programu svařování méně než pět.

Krok 1, zda použít rozšířenou osu, pokud je rozšířená osa použita, je třeba nakonfigurovat související souřadnicový systém rozšířené osy a povolit rozšířenou osu.



Obrázek 4.9-21 Rozšířená konfigurace osy

Krok 2: Kalibrujte počáteční bod, bezpečnostní bod počátečního bodu, koncový bod a bezpečnostní bod koncového bodu. Pokud je v prvním kroku vybrána rozšířená osa, načte se funkce pohybu rozšířené osy, která bude spolupracovat s kalibrací oMezních bodů.





V kroku 3 zvolte, zda je potřeba laser, a pokud ano, upravte parametry příkazu pro polohování laserem.

**Whether	use	laser	sensor?
		1010.01	

Cancel		No	Yes
Search	Time		ms
Search	Length		mm
Search	Speed		%
Search	Dist		T



Krok 4: Vyberte, zda je vyžadováno svařování tkaním, a pokud je svařování tkaním vyžadováno, je třeba upravit příslušné parametry svařování tkaním.

**Whether use	weave?		
Weave	ID	0 •	
Weave	Туре	平面三角波摆:▼	]
Weave Freq	uency	0.000000	Hz
Weave F	Range	0.000000	mm
Left Stay	Time	0.000000	ms
Right Stay	Time	0.000000	ms
Cancel		No Y	′es



Krok 5, pojmenujte program a automaticky jej otevřete v rozhraní pro výuku programů.



#### Obrázek 4.9-25 Uložení programu

Kliknutím na položku "Arc Welding" (Obloukové svařování) v části "Weldment Shape" (Tvar svařence) vstupte do rozhraní pro navádění k obloukovému svařování. Na základě základního nastavení robota můžeme rychle vygenerovat výukový program svařování pomocí dvou jednoduchých kroků. Zahrnuje především následující dva kroky.

Krok 1: Kalibrujte počáteční bod, bezpečnostní bod počátečního bodu, bod přechodu oblouku, koncový bod a bezpečnostní bod koncového bodu.



Obrázek 4.9-26 Kalibrační bod

Krok 2: Pojmenujte program a automaticky jej otevřete v rozhraní pro výuku programů.



Obrázek 4.9-27 Uložení programu

Kliknutím na "Multi-layer multi-pass welding" (Vícevrstvé víceprůchodové svařování) v části "Weldment shape" (Tvar svařence) vstupte do rozhraní pro navádění k vícevrstvému víceprůchodovému svařování. Po dokončení konfigurace různých základních nastavení robota můžeme rychle vygenerovat výukový program svařování pomocí čtyř jednoduchých kroků. Zahrnuje především následujících pět kroků.

Krok 1. Nastavte první skupinu bodů podle pokynů, a to bod svařování, bod X+, bod Z+ a bezpečnostní bod.

### Multi layer and multi pass welding





Krok 2, druhá skupina nastavení bodu, můžete nastavit typ bodu dráhy, podporovat přímku a obloukovou dráhu, včetně bodu svařování, bodu X+ a bodu Z+.



Obrázek 4.9-29 Druhá sada nastavení bodů

Krok 3: Po nastavení všech skupinových bodů klikněte na tlačítko "Dokončit", čímž vstoupíte na stránku s funkcí nastavení posunu jednotlivých svarových paprsků, a postupně nastavte posun požadovaného svarového paprsku. Rozhraní je znázorněno na obrázku níže.

- <b>≺</b> ≡	and the second second	2	0	•	9		Stopped 10011 wobj1 exaxis1	100 \$		33	3 6
🗘 Initial Setup <	D Part Snape -	Multi laye	r and mul	ti pasa we	elding		Acard Same That Mann Mann (	Robot P	000		C
∃ Trach Simul € I⊷ Status Info - €	Line Seam Arc Seam			G	t.		Speed 100 10 100	21.10 24.10	11.1	D N	Join
8 Autiliary V	Mitt Seam		ć	11.			Length Line 50 *	X :0 RX-0	V is Red	t R2	70 10
Professional Weld	<ul> <li>Fortune Str.</li> </ul>	Count	01/01	X(mm)	Z(mm)	-	⊖ -o-	-			1
Security settings	81 Conto	3	a)	0	0	0	○	Py in Tailo	Fr : 0	11	0
System Setting		(# ) ·	R.(	0	0	0	🗢 o 😲 📍	ALLSIN			
		3	D : .	0	0	0	○ ○				Line-N
		4	10	4	0	0	G0- C	There is a			
		5	9	0	0	0	🗢 🛟 🔤 🛛	004	001	003	Ctrill Dog
		4	0	6	6	6	Deela	004	001	000	007 000
		7	8	0	0	0	Port.	00	का	0.0	01
			10	0	0	0	Acres 1	Dis	0.05	0.04	0.0

Obrázek 4.9-30 Nastavení posunu svařovací fazety

Krok 4: Po nastavení všech požadovaných parametrů svařence klikněte na tlačítko "Finish", čímž přejdete na stránku pro generování programu, zadejte název souboru a může být vygenerován vícevrstvý víceprůchodový svařovací program, který pak uživatel může otevřít ve výuce programu Program je odladěn a rozhraní je znázorněno na obrázku níže.



Obrázek 4.9-31 Uložení programu

### 1.3.5.9.13 Bezpečné nastavení rychlosti

V části "DI Configuration" (Konfigurace DI) v části "Robot Settings" (Nastavení robota) klikněte na různá rozevírací pole DI a nakonfigurujte režim redukce (úroveň 1, úroveň 2, úroveň 3).

V režimech redukce první a druhé úrovně lze konfigurovat společnou rychlost a rychlost koncového TCP a v režimu redukce třetí úrovně je možné zastavit bez konfigurace rychlosti.



a ≡	-	()	Stopped toolcoord0 wobj0 exaxis0 10 <b>b</b> \Lambda 🔞 🕫 🔗
Initialize	World coord.		Apply Operation&Status 360° Free Mounting Fixed Mounting
Robot Settings	<ul> <li>Tool coord.</li> <li>Ext. tool coord.</li> </ul>	End Input DI0 nothing V DI1	ofbing v Fixed 10 TFD FT RCM
Teaching <	Workpiece co	DID Active at low I 🗸 DI1	Apply         Speed         100         %         J1:55/209         J2:56/209         J2:58/201         J3:59/41           clive at low 1v         Acceleration         180         759/2         180         19:52
-lµ- Status < BB Auxiliary <	Collision level		Apply         Threshold         30         *         Threshold         30         *         *         *         : <511 813 V : :455 319 Z : :554834
★ Settings	<ul> <li>End load</li> <li>Friction comp</li> </ul>	Security Stop Policy suspend V	Apply         J1         →         →         →         FT           J2         →         →         →         00.51         Tot         F1:0.000         F2:0.000
	Speed scaling	Reduction Mode Configuration	J3 ← − ← 59.81 Tx : 9.000 Ty : 9.000 Tz : 9.000 ACI State: 1
	IO titering	Reduction Mode         Level 1 reduction ✓           J1         30         Level 1 reduction mode           Level 2 reduction mode         Level 2 reduction mode	J4 🗢 🕀
	DO config	12 30 */s	J5
	Al config	J4 30 */s	
	Gonfig import	J5 30 */s	Prefix CC0 C01 C02 C03 CC4 C05 C06 C07
		TCP 300 mm/s	Asply         Dia         Dia </th

Obrázek 4.9-32 Bezpečné nastavení rychlosti

### 1.3.5.9.14 Konfigurace bezpečnostní stěny

V nabídce "Security Settings" (Nastavení zabezpečení) v "Auxiliary Application" (Pomocná aplikace) klikněte na "Security Wall Configuration" (Konfigurace bezpečnostní stěny), čímž vstoupíte do rozhraní funkce konfigurace bezpečnostní stěny.

Konfigurace bezpečnostní zdi Kliknutím na tlačítko povolit povolíte příslušnou bezpečnostní zeď. Pokud
není bezpečnostní stěna nakonfigurována s bezpečnostním rozsahem, zobrazí se chybové hlášení. Klikněte na
rozevírací pole, vyberte bezpečnostní stěnu, kterou chcete nastavit, a automaticky vyvolejte bezpečnostní
vzdálenost (nelze ji nastavit, výchozí hodnota je 0) a poté kliknutím na tlačítko "Nastavení" úspěšně
nastavte.



Obrázek 4.9-33 Konfigurace bezpečnostní zdi

 Konfigurace referenčního bodu bezpečnostní stěnyPo výběru bezpečnostní stěny lze nastavit čtyři referenční body. První tři body jsou rovinné referenční body, které slouží k potvrzení roviny nastavené bezpečnostní stěny. Čtvrtý bod je referenční bod bezpečnostního rozsahu, který se používá k potvrzení bezpečnostního

## <u>M, verze 1.0.0</u>

rozsahu nastavené bezpečnostní stěny.

Důležité: Pokud je referenční bod úspěšně nastaven, svítí zelená kontrolka. V opačném případě svítí žlutá kontrolka. Svítí zeleně, dokud není referenční bod úspěšně nastaven. Pokud jsou všechny čtyři referenční body úspěšně nastaveny, lze vypočítat bezpečnostní rozsah a po úspěšném výpočtu se stav parametru bodu bezpečnostního rozsahu vrátí na výchozí hodnotu.

a ≡			Stopped toolcoord0	wobj0 exaxis0 10 🖌 🔬 🔞 🔁 ⊗
🐵 initialize <	G Safe speed s.	Plane3	Operation&Status 360*	Free Mounting Fixed Mounting
🗐 Teaching <	Security Stop	Ptane4	Joint Base Tool Wool Move	Robot Pose
J. Status	Security wait	Plane5	Easts 10 TPD FT RCM	
88 Auxiliary 🗸	Security dae	Plane6	Speed 100 %	Joints J1 1-138.895 32 - 79.454 J3 175.005 J4 1-107.968 JE - 91.988 J6 199.728
Robot Body Welding Library		Piane/	Threshold 30 *	TCP X 1390.306 Y 1471.521 Z 1546.406 RX1-161.405 RY1 12.644 RZ1-65.92
Security Settings		Security wall configuration Choose Plano1 v Sate dist 10.000000 mm	Set 15 00 000 000 000 000 000 000 000 000 0	Tox Fx:0.000 Fy:0.000 Fz:0.000 Tx:0.000 Tz:0.000 Tz:0.000 Act State: 1
		Plane reterence point	Set 34	Line-Num
		Reference point2	Set 18	CtriBox
		Reference point3	Set Jos 200	D06 D01 D02 D03 D04 D06 D06 D07 C09 C01 C02 C03
		Safe Range Reference Point Reference point4	Profix Set Point name Add	
			Sensor v	Ciú Ci1 Ci2 Ci3 Ci4 Ci5 Ci6 Ci7 Ci4 Ci5 Ci6 Ci7

Obrázek 4.9-34 Nastavení referenčního bodu bezpečného rozsahu

 Použít efekty: Úspěšně nakonfigurovaná bezpečnostní zeď je povolena. Přetáhněte robota, pokud je TCP na konci robota v nastaveném bezpečnostním rozsahu, je systém v pořádku. Pokud je mimo nastavený bezpečnostní rozsah, zobrazí se chybové hlášení.

a ≡			Stopped toolcoord0 wobj0 exaxis0 10 🖢 🗚 🔞 😰 ⊗
Initialize        Imitalize        Imitalize <td< th=""><th><ul> <li>Sale speed s.</li> <li>Security Stop.</li> <li>Security wall.</li> <li>Security tae.</li> </ul></th><th>Constant on figuration     Security wall configuration     Security wall configuration     Plane1     Plane2     Plane3     Plane3     Plane5     Plane6     Plane6     Plane8     Constant on     Security wall configuration     Security wall     Securi</th><th>Stopped         toolcoord0         wobj0         exaxis0         10         Image: Control of the security well           OperationS Status         300 "F"         Twaning/Access to the security well         Control of the security well           Initiation of the security of the se</th></td<>	<ul> <li>Sale speed s.</li> <li>Security Stop.</li> <li>Security wall.</li> <li>Security tae.</li> </ul>	Constant on figuration     Security wall configuration     Security wall configuration     Plane1     Plane2     Plane3     Plane3     Plane5     Plane6     Plane6     Plane8     Constant on     Security wall configuration     Security wall     Securi	Stopped         toolcoord0         wobj0         exaxis0         10         Image: Control of the security well           OperationS Status         300 "F"         Twaning/Access to the security well         Control of the security well           Initiation of the security of the se
		Reference point2	Sensor Ci0 Ci1 Ci2 Ci3 Ci4 Ci5 Ci5 Ci7

Obrázek 4.9-35 Obrázek efektu po úspěšném nastavení rozsahu zabezpečení

## 1.3.5.9.15 Bezpečnostní démon

V nabídce "Security Settings" (Nastavení zabezpečení) v části "Auxiliary Apps" (Pomocné aplikace) klikněte na položku "Security Background Program" (Program na pozadí zabezpečení), čímž vstoupíte do rozhraní funkce programu na pozadí zabezpečení.

Uživatel kliknutím na tlačítko "funkce povolena" otevře nebo zavře nastavení bezpečnostního démona. Vyberte "Neočekávaná situace" a "Program na pozadí" a kliknutím na tlačítko "Nastavení" nakonfigurujte parametry logiky zpracování neočekávaných situací.

Povolte bezpečnostní program na pozadí a nastavte neočekávanou scénu a program na pozadí. Když uživatel spustí program a neočekávaná situace odpovídá nastavené neočekávané situaci, robot spustí odpovídající program na pozadí, který bude hrát roli bezpečnostní ochrany.



Obrázek 4.9-36 Bezpečnostní démon

## 1.3.5.10 Nastavení systému

### 1.3.5.10.1 Obecná nastavení

Klikněte na položku "System Settings" (Nastavení systému) na levém panelu nabídek a kliknutím na položku "General Settings" (Obecná nastavení) na druhém panelu nabídek vstupte do rozhraní obecných nastavení. Obecné nastavení může aktualizovat systémový čas robota podle aktuálního času počítače, takže čas oMěření obsahu protokolu je přesný.

V nastavení sítě lze nastavit IP adresu řadiče, masku podsítě, výchozí bránu, server DNS a IP adresu učícího přívěsku (tato IP adresa je platná při použití našeho učícího přívěsku M -HMI a musí být použita při použití učícího přívěsku M -HMI Konfigurace učícího přívěsku je povolena), což je pro zákazníky výhodné při používání scény.

Zákazníci si mohou podle svých potřeb zvolit jazyk výuky jako čínštinu, angličtinu, m enšinu nebo japonštinu. Jazyk si navíc může uživatel přizpůsobit. Uživatel může přeložit exportovaný čínský jazykový soubor, importovat přeložený jazykový soubor a poté zvolit použití importovaného jazykového souboru.

Uživatelé mohou nastavit počet uchovávaných protokolů a importovat a exportovat konfigurační soubor systému. Maximální počet oMetained logs je 30 a konfigurační soubor systému zaznamenává nastavenou hodnotu.

Obnovení továrního nastavení v rámci obnovení systému může vymazat uživatelská data a obnovit tovární nastavení robota.

Funkce generování protokolu podřízené stanice a exportu protokolu řídicí jednotky slouží ke stažení některých

důležitých souborů se stavovými nebo chybovými záznamy řídicí jednotky, což je vhodné pro řešení problémů s robotem.

### 1.3.5.10.1.1 Nastavení sítě

≡ &			Stopped toolcoord0 wobj0 exaxis0 10 <b>b A</b> 🔞 😰 ⊗
Image: Arrow of the section of th	Q     General       Image: Account     General	Image: Provide state         Image: Provide state           Ethernet0         Image: Provide state         Image: Provide state           Submet Mask         255:255:255:0         Image: Provide state           Default Gataway         192:168:58:1         Image: Provide state           DNS Server         192:168:58:1         Image: Provide state           Ethernet1         Image: Provide state         Image: Provide state           IP         192:168:58:2         Image: Provide state           Submet Mask         255:255:255:0         Image: Provide state	Stopped toolcoord0 wobj0 exaxis0 10 L A R r2 S
		Default Gateriagy 192, 168, 58, 1 DNS Servar 192, 168, 58, 1 DNS Servar 192, 168, 58, 1 Teach Pendant Access IP WebAPP Ethernet @Etherne WobRecovery @Ethernet) Etherne tet Network	Affiel box nobleost

Obrázek 4.10-1 Schéma nastavení sítě

- Set ethernetZadejte IP adresu síťové karty, která má komunikovat, masku podsítě (spojenou s IP adresou, automaticky vyplněnou), výchozí bránu a server DNS. Tovární výchozí IP síťové karty 0 síťového portu: 192.168.57.2, výchozí IP síťové karty 1: 192.168.58.2.
- Učitelský přívěsek povolenŘídí, zda je povolen učitelský přívěsek. Ve výchozím nastavení je učící přívěsek vypnutý a zařízení nelze ovládat pomocí učícího přívěsku. Kliknutím na tlačítko posuvného přepínače povolíte ovládání zařízení pomocí učícího přívěsku.
- **Přístup IPZvolt**e síťovou kartu přidruženou k WebAPP a WebRecovery. Když je zapnutý přívěsek pro učení, WebAPP ve výchozím nastavení vybere síťovou kartu 1 a síťová karta 0 není volitelná.
- Nastavit síť Klikněte na tlačítko "Nastavit síť", čímž se zobrazí výzva, že probíhá konfigurace. Po dokončení konfigurace je třeba zařízení restartovat.

### 1.3.5.10.2 Nastavení účtu Nastavení účtu

Klepnutím na položku Nastavení účtu na sekundárním panelu nabídek vstoupíte do rozhraní Nastavení účtu. Funkce správy účtu jsou dostupné pouze správcům. Funkce jsou rozděleny do následujících tří modulů:

### 1.3.5.10.2.1 Správa uživatelů

Stránka pro správu uživatelů, slouží k ukládání informací o uživatelích, můžete přidat ID uživatele, funkci atd. Uživatel se může přihlásit zadáním stávajícího uživatelského jména a hesla v seznamu uživatelů.

ā ≡		4					Stop	ped toolcoord	0 wobj0	exaxis0	10	3	20
Initialize <	<b>O</b> General	Account m	anagement > Us	er Managemen	ŧ								
Teaching K	🖪 Account 🖂	User mana	gement lists										
-lµ. Status <	User Management	<b>Q</b> S	earch				Add	Delete					
🖽 Auxiliary 🗸	Authority	0	Work ID	Name	Password	Function name	Function code	operation					
V Settings	Management		111	admin	*****	管理员	1	17 E					
X secondo	Import/Export	0	222	MEengine er		MET #205	2	2 0					
	About		333	PEengine er		PE、PQE工程 版	3	20					
	Custom	0	444	technician		技术员和短组长	4	28					
	Maintain	0	555	operator		操作员	5	2 8					
			666	monitor	*******	监视	6	28					
			mmmm	test123		ME工程序	2	28					
		D	1234567890	使 型	******	示敵管理员	66	28					

## Obrázek 4.10-2 Správa uživatelů

• Přidání uživatelůKlikněte na tlačítko "Přidat", zadejte číslo úlohy, jméno, heslo a vyberte funkci.

Důležité: Číslo úlohy může být až desetimístné celé číslo, číslo úlohy a heslo jsou jednoznačně kontrolovány a heslo je zobrazeno Braillovým písmem. Po úspěšném přidání uživatele můžete zadat jméno a heslo a znovu se přihlásit.

-1 =					stopped	tooicoordu	wobju	exaxis0	10	A	0	<b>Z</b> (8
) Initialize <	<b>O</b> General	Account management > U	ser Management > A	dd								
Teaching <	🖪 Account 🗸	Add user information										
. Status 🗸	User Management	Work ID	10									
g Auxiliary <	Authority	Name										
Settings	Management Import/Export	Password										
	BJ Plugn	Function hame		~								
	< About		Cancel	Save								
	E Custom											
	<ul> <li>Maintain</li> </ul>											

### Obrázek 4.10-3 Přidání uživatelů

• Úprava uživatelůKdyž je k dispozici seznam uživatelů, klikněte na tlačítko "Upravit" vpravo, číslo úlohy a jméno nelze upravit, ale heslo a funkce lze upravit a heslo je třeba také jednoznačně ověřit.

a ≡	5		Stopped	toolcoord0	wobj0	exaxis0	10	4	3	20
⊕ initialize <	Q General	Account management > User Management > Add								
Teaching (	🖪 Account 🖂	Add user information								
	User Management	Work ID 921739812								
🔠 Auxiliary 🗸	Authority	Name Jack								
X Settings	Management Import/Export Plugin About Custom	Pissword *** Function name								

Obrázek 4.10-4 Úpravy uživatelů

• Mazání uživatelůZpůsoby mazání se dělí na jednorázové mazání a hromadné mazání.

1. Klikněte na jediné tlačítko "Smazat" na pravé straně seznamu a zobrazí se výzva "Pro potvrzení smazání klikněte znovu na tlačítko smazat" a pro úspěšné smazání znovu klikněte na seznam.

2. Klikněte na zaškrtávací políčko vlevo, vyberte uživatele, kteří mají být odstraněni, a poté dvakrát klikněte na tlačítko dávky "Odstranit" v horní části seznamu pro odstranění.

Initialize 🔇	O <sub>6</sub> General	Account ma	inagement > Use	er Managemen	t						
Teaching <	🖸 Account 🤟	User manag	ement lists				Ē				
Status 🗸	User	<b>Q</b> 56	arch				Add	Delete			
Auxiliary 🗸	Authority	23	Work ID	Name	Password	Function name	Function code	operation			
Settings	Management		111	admin		管理员	1	17 B			
	Import/Export	53	222	MEengine er		ME工程团	2	6			
	About	53	333	PEengine er		PE、PQE工程 师	3	2 🖻			
	Custom	23	444	technician		技术员&把组长	4	28			
	Maintain	52	565	operator		操作员	5	2 8			
		53	666	monitor		监视	6	2 8			
		52	111111111	test123		ME工程师	2	28			
		2	1234567890	赵嬰		示敵管理员	66	2 8			

Obrázek 4.10-5 Odstranění uživatelů

## 1.3.5.10.2.2 Řízení úřadu

**Důležité:** Výchozí data funkce (kód funkce 1-6) nelze vymazat a kód funkce nelze změnit, lze však změnit název a popis funkce a nastavit autoritu funkce.



Obrázek 4.10-6 Správa autorit

Ve výchozím nastavení je k dispozici šest funkcí, správci nemají žádná funkční omezení, operátoři a monitoři mohou používat malý počet funkcí, inženýři ME, inženýři PE&PQE a technici a vedoucí týmů mají některá funkční omezení, správci nemají žádná funkční omezení, konkrétní výchozí oprávnění jsou uvedena v následující tabulce:

Důležité: Výchozí oprávnění lze změnit

Tabulka 4.10-1 Podrobnosti o povolení

Category	Permission item	Administrator	Engineer 1(ME)	Engineer2(PE,PQE)	Fechnician & Team Leade	Operator	Monitor
	Robot Setting	YES	YES	YES	YES	YES	NO.
	Peripheral Configuration	YES	YES	YES	YES	NO	NO
	Teach Programming	VFS.	VES	VES	VES	WS.	VES
	Granbural Programming	VES	YES	VPC	VER	WE	VER
	Teaching Management	VPS.	VES	VPL	VES	47.0	VES
	Sutem Ion	VES	YES	VES	VES	YES	VES
	Status Query	YES	YES	YES	YES	NO	NO
Accessible view	Accellary Application	Ws	VES	YES	YES	NO	NO
	Welding Expert Library	YES	YES	YES	NO	ND	NO
	Security Setting	YES	VES	YES	NO	NO	NO
	General Setting	YES	YES	YES	YES	NO	NO.
	Plugin Setting	YES	VES	YES	NO	NO	NO
	Custom Information	YES	YES	YES	NO	NO	NO
	Maintenance Mode	YES 1	NO	NO	NO	NO.	NO
	Start/Stop/Pause & Resume	Y\$S	YES	YES	YES	YES	NO
	Speed Scaling	YES	VES	VES	YES	NO	NO
	Manual/Auto Switch	YES	YES	YES	YES	YES	NO
	Drag Mode Switch	YES	YES	YES	YES	ND	NO:
	Free Mounting	YES	VES	YES	NO	NO	ND
	Fixed Mounting	YES	YES	YES	YES	NO	NO
	Joint	YES	VES	YES	YES	725	NO.
	8250	YES	YES	YES	YES	TES	NU:
	3001	YES	YES	YES	YES	YES	NU
	Wobj	TLS	YES	YES.	YES	nts-	NU.
	NOVE CONTRACTOR	110	VER	Up:	VEE	NO.	NO
	1/0	VEC	VES	VIC.	VEE	VEC	NO
	190	VEC	ALCO	Vic	VEE	Vite 1	NO
	FT	Ws	YES	yre:	VER	NO	NO
	RCM	VES	YES	YES	YES	ND	NO
	Teaching Point Recording	VES	VES	YES	YES	YES	NO
	Sensor Paint Recording	YPS	YES	YPS	YES	YES	NO
	Base Coordinate System 3D Dinnlaw	YES	Vyes	YES	YES	YES	YPS
	Tool Coordinate System 3D Display	YES	YES	YES	YES	YFS	VES
	Workpiece Coordinate System 3D Dania	YES	YES	VES	YES	YES	YPS
	Extended Axis Coordinate System 3D Dia	VES	VES	VES	YES	YES	VES
	Trajectory Drawing	VES	VES	VES	YES	YES	NO
	Import Tool Model	YES	YES	YES	VES	ND	NO
	World Coordinate System	YES	NO	NO	NO	NO	NO
	Tool Coordinate System	YES	YES	YES	YES	YES	NO.
	External Tool Coordinate System	YES	YES	YES	YES	YES	NO:
	Workpiece Coordinate System	YES	VES	YES	YES	YES	NO.
	Extended Axis Coordinate System	YES	YES	YES	YES	125	NO
	Collision Level	YES	VES	VES	YES	YES	NO
	Soft Lenit	YES	YES	YES	NO	NO	NO
	End Load	YES	YES	YES	YES	YES	NO
	Friction Compensation	YES	YES	YES	YES	NO	NO
	Speed Scaling	YES	YES	YES	YES	YES	NO.
	VO Filtering	YES	VES	YES	YES	NO	NO
	Di Configuration	YES	YES	YES	YE5	NO	NO
	Do Configuration	YES .	YE5	YES	YEB	ND	NO.
	Ai Configuration	YES	YES	YES	YES	NO	NO
	File Import And Export	YES	YES	YES	YES	NO	NO
	Peripheral Configuration	YES	YES	YES	YES	NO	NO
	Spray Gun Configuration	YES	YES	YES	YES	NO	NO
	Welder Configuration	YES	71:5	YIS	YES	ND	NO
Operable function	Sensor Tracking	YES	YES	YES	YES	NO	NO
	Extended Axis	YES	YES	YES	YES	NO	NO.
	Conveyor Tracking	YES	YE5	YES	YE5	NO	NO.
	Track Pose	YES	YES	YES	YES	NO	NO
	Torque System	YES	YES	YES	YES	NO	NO.
	Health Care System	YES	VÉS	YES	YES	NO	NO
	Palletizing System	YES	YES	YES	YES	NO	NO
	Polishing Device	rts	YES	YIS	YES	ND	NO
	emport -	YES	YES	YES	YES	NO	NO
	txport	115	785	YES	YES	NO	NO
	modify Debug	125	YES	YES	NO	NO	NO
	Deece	165	TES	185	YES	Dig	NO.
	Robotic Correction	VEC .	100	115	TEN .	140	NO.
	Encoder Configuration	123	100	100	80	NO-	NO:
	System Upprade	WS.	NO	NO	NO	NO	NC
	Data Back o	VES	YES	YES	YES	NO	NO
	105 Data Recording	YES	- WES	YES	YES	NO	NO
	Teach Point Configuration	WE5	YES	YES	YES	NO	NO
	Matrix Move	VES	YES	YES	YES	NO	NO
	Starting Point	YES	VES	YES	YES	NO	NO
	Interference Zone Configuration	YES	VES	YES	YES	NO	NO
	End Led Configuration	YES	YES	YES	YES	ND	NO
	Custom Protocol	YES	NO	NO	NO	NO	NO
	Peripheral Protocol	YES	YES	YES	NO	NO	NO
	Main Program Configuration	YES	YE5	YES	YES	NO	NO
	Drag Lock	YES	VES	YES	YES	NO	NO
	Smart Tool	YES	YES	YES	YES	NO	NO
	Multiple Interlerence Zone Configuration	YES	VES	YES	YES	NO	NO
	Safe Speed Setting	YES	YES	YES	NO	NO	NO.
	Safe Stop Setting	YES	YES	YES	NO	NO	NO
	Safety Plane	YES	VES	YES	NO	NO	NO
	Security daemon	YES	YES	YES	NO	NO	NO
	Time Setting	YES	YES	YES	YES	NO	NO
	Network Setting	YES	YE5	YES	YE5	NÐ	NO
	Teach Fendant Setting	YES	YES	YES	YES	NO.	NO
	System Language	YES	YES	YES	YES	NO	NO
	Log Management	YES	YES	YES	YES	ND	NO
	Logaut Timeout	YES	YES	YES	YES	NO	NO
	System file Export	YES	YES	YES	YES	NO	NO
	Upload Information Package	YES	NO	NO	NO	NO	NO
	Modify Robot Type Name	YES	NO	NO	NO	NO	NO

• **Přidat funkci**: Klikněte na tlačítko "Přidat", zadejte kód funkce, název funkce a popis funkce, klikněte na tlačítko "Uložit" a po úspěchu se vraťte na stránku se seznamem. Mezi nimi může být kód funkce pouze celé číslo větší než 0 a nesmí být stejný jako kód existující funkce a všechny vstupní položky jsou povinné.



## Obrázek 4.10-7 Funkce Přidat

• Upravte název a popis funkce: Klepnutím na ikonu "Upravit" na operačním panelu tabulky můžete upravit název a popis funkce aktuální funkce. Po dokončení úpravy klikněte na tlačítko "Save" (Uložit) níže pro potvrzení úpravy.

ā ≡	1	<b>(4</b> )	€	•						St	opped	toolcoord0	wabj0	exaxis0	10	bi I	<b>A</b>	5	88
∰ Initialize ≮	🥵 General	Account management - A	uthority	assignment > Ed	R.														
E Institut	🖪 Account -	Edit function information																	
	User	Function code	1																
Ji Diatus 🤇	Management	Function name	管理员				33												
BB Autiliary <	Authority	Job description	製用品																
* Settings	InnortExport						6												
	65				- 64	encel 0	Seve												
	es rign																		
	< About																		
	Custom																		
	<ul> <li>Maritan</li> </ul>																		

## Obrázek 4.10-8 Úprava názvu a popisu funkce

 Nastavení oprávnění k funkcím: Klepnutím na ikonu "Nastavení" na ovládacím panelu tabulky nastavte oprávnění aktuální funkce. Po nastavení klikněte na tlačítko "Uložit" níže pro potvrzení nastavení.

<i>7</i> 3 ≡		<b>(</b> )				Stopped toolcoord0 wobj0	exaxis0 10 🖌 🛆 🖄 🛫 🛞
∰ Initialize ∢	9, General	Account management > Au	athonity assignment > Set				â
E Institut (	Account -	Set functional permissions					
A Status (	User	Current function					
99 Autilians	Management	Accessible view	Robol Setting	Cirera and	pheral Configuration	Teach Programming	Graphical Programming
A tamiana	Management	-	Calleaching Manageria	eur Casia	cem Log	Contra Guery	
× secondar	ImportExport		Caveding Expert Libs	ay Viseo	uniy setting	Ceneral Setting	Alendia segui
	Pagen	Actionable autorations		Centre & Desime	Fiscart Scator	Manual Area States	Cline tande States
	< About	Advisor Innovate	Clothe swith	Characteria and a weather	Calebra Scaling	Contraction Contract	Carlos more samo
	E Custom		Robot operation	Free Mounting	Fixed Mounting	2 Joint	Base
	Marrian			706	Mod	Move	2 EANS
				<b>MID</b>	TPO	<b>⊠</b> FT	RCM
				Teaching Point Recording	Sensor Point Recording	Base Coordinate System 30 Displa	Tool Coordinate System 3D Display
				Workpiece Coordinate System	T 3D	iyelem Minajectory Drawing	Import Tool Madel
			Robot Setting	World Coordinate System	Tool Coordinate System	External Tool Coordinate System	Workpiece Coordinate System
				Extended Axis Coordinate Sys	stem	Sot Limit	End Load
				Friktion Compensation	Speed Scaling	Cito Fiteing	Di Configuration
				Do Configuration	Al Configuration	File Import And Export	
			Perpheral Configuration	Perpheral Configuration	Spray Gun Configuration	Weider Configuration	Sensor Tracking
				Extended Axis	Conveyor Tracking	Track Pose	Torque System
				Health Cale System	Palletizing System		
			Teaching Nanagement	<b>M</b> import	Export	Mosity	Delete
● Initialize 《 □ Teoching 《 → Stafus 《 昭 Austram 《 ▼ Seminge	<ul> <li>General</li> <li>Account of Management</li> <li>Ler Management</li> <li>Addutothy Management</li> <li>About</li> <li>Flagn</li> <li>About</li> <li>Custom</li> <li>Monten</li> </ul>		Procest Setting Perphereil Configuration Tracting Management Tracting Management Stylem Lög Ausstary Application Secordy Setting Greenedi Setting	Nempere Continue Bytem     Nempere Continue Bytem     Neme Continues Bytem     Deconfiguration     De	1.50. Elemented Ave Coordnete 8 for Coordnete 8 (Confision Leve Speed Scaling (A) Configuration (String Gun Configuration (Configuration) (Configuration (Configuration) (Configuration (Configuration) (Configuration) (Configuration)	ystem	Clempert Tool Medel Westpiece Coordinate System Clend Load Cloid Carifigination Cleaning Fracting Cleaning Fracting Cleaning Frant Cleaning Frant Clea
			Custom Information	Upload Information Package	Modify Robot Type Name	Teaching Program Encryption	

Obrázek 4.10-9 Nastavení oprávnění funkcí

• Funkce Odstranit: Pokud aktuální funkci nepoužívá žádný uživatel, lze ji smazat, v opačném případě ji smazat nelze.

a ≡		4				Stopped	toolcoord0	wobj0	exaxis0	10	<b>A</b> 8	9 17	10
∯ Initialize ∢	9, General	Account management	nt - Authority assignmen										
El Teaching 🖌	🖪 Account -	Functional authority m	lanagement										
A-Status K	User Managament	Ridd											
88 Austiany <	Authority	Function	Function name	Job description	Operate								
* Settings	ImportExport	1	管理员	管理员	Ø								
	93 Plugn	2	ME工程师	MEINER	Ø								
	< About	3	PE, POLIZIE	PE, PORING	C8 🗘								
	E Custom		技术员会现记任	技术员638组长	20								
	a Marntain	5	操作员	操作品	Ø								
		6	后现	35%	₿\$								
		9	牌法工程研	教法工程师	60								

## Obrázek 4.10-10 Funkce Odstranit

## 1.3.5.10.2.3 Import/export

5 ≣		Stopped   toolcoord0   wobj0   exavis0   10   🖬   🛦   🕲 🥳   ⊗
→     Initialize     <       →     Initialize     <       →     Status     <       →     Status     <       ★     Status     <       ★     Sentings	Central   Account management   Account management   Account management   Managemont   Activity   Managemont   Activity   Managemont   Coston   Octoon   Mantain	Stopped toolcoord0 wobj0 exaxis0 10 🖬 \Lambda 🗑 🕵 🛞

# Obrázek 4.10-11 Import/export nastavení účtu

- Dovoz: Klepnutím na tlačítko "Importovat" můžete dávkově importovat data pro správu uživatelů a práv.
- Export: Kliknutím na tlačítko "Exportovat" můžete dávkově exportovat data správy uživatelů a práv.

## 1.3.5.10.3 O stránkách

Klepnutím na tlačítko O aplikaci na sekundárním panelu nabídek vstoupíte do rozhraní O aplikaci. Na této stránce se zobrazí model a sériové číslo robota, verze webu a verze řídicí jednotky používané robotem, verze hardwaru a verze firmwaru.

	🏂   🕟 💿 🕕 Stopped toolcoord0 wobj0 exaxis0 0 🖢 🛦 🔞 💆	8
🕼 Initialize 🧹 🤷 General	About	. Î
E Teaching	Version information	
B Plugin	Robot model FR5	
About	Web version	
Hand Auxiliary Custom	Controller version	- 1
X Settings	Hardware version Carrier plate T	
	Firmware version Carrier plate	
	slave chip Carrier plate 🔻	- 1
	Stave Profile Carrier plate • id:	
	code:	
	edition:	
	联系我们	
	地址 江苏省苏州市高新区村园路209号2号楼	
	地址 山东省淄博市高新区等资路6888号	
	电话 0512-68562005	
	即//中 friech@friech.fr	
	网让 www.fitech fr	÷

Obrázek 4.10-12 O schématech

#### 1.3.5.10.4 Vlastní informace

Kliknutím na vlastní informace na sekundárním panelu nabídky vstoupíte do rozhraní vlastních informací. Funkce vlastních informací mohou používat pouze správci. Na tuto stránku lze nahrát balíčky informací o uživateli, vlastní modely robotů a nastavit stav šifrování výukového programu.

ā ≡		6 I I II	)	Stopped	toolcoord0 wobj0 exa	xis0 10 🖌 🕭 🕱 😤 🛞
<ul> <li>Initialize</li> <li>✓</li> <l< th=""><th><ul> <li>General</li> <li>Account</li> <li>Plugin</li> <li>About</li> <li>Custom</li> <li>Maintain</li> </ul></th><th>User Information Configuration User Information package upload Upload Information package Biggit Attached Please enter the model Apply</th><th>选择任何文件</th><th></th><th></th><th></th></l<></ul>	<ul> <li>General</li> <li>Account</li> <li>Plugin</li> <li>About</li> <li>Custom</li> <li>Maintain</li> </ul>	User Information Configuration User Information package upload Upload Information package Biggit Attached Please enter the model Apply	选择任何文件			
		Demonstration of teaching procedures Name Search				
		Program name	Not encrypted	First-level encryption	Second-level encryption	First-level encryption
		201.lua		<b>6</b> D	<b>6</b> D	function Can read
		202.kua		60	<b>6</b>	C Edit C Save
		203.kua		60	<b>6</b> 0	C Export C Delete
		25 Jua		<b>6</b> 0	60	O Rename O Save As
		26 Jua		<b>(</b> )	(E)	<ul> <li>Import progr</li> </ul>

Obrázek 4.10-13 Schéma vlastních informací

## 1.3.5.10.4.1 Konfigurace rozsahu parametrů

Konfigurace rozsahu parametrů, pouze správce může nastavit rozsah parametrů a parametry ostatních oprávněných členů mohou být nastaveny pouze v rámci rozsahu parametrů nastaveného správcem.

Parametry lze nastavit dvěma způsoby: přetažením posuvníku a ručním zadáním.

**Důležité:** Maximální hodnota rozsahu parametrů musí být větší než minimální hodnota. Po 3 sekundách od úspěšného nastavení rozsahu parametrů dojde k automatickému přeskoku na přihlašovací stránku a je třeba se znovu přihlásit.



Obrázek 4.10-14 Schéma konfigurace rozsahu parametrů

# 1.3.6 Periferní

## 1.3.6.1 Konfigurace periferií chapadla

### 1.3.6.1.1 Kroky výuky programu Gripper

Krok1V rozhraní konfigurace uživatelských periferií **vyberte** tlačítko "End Peripheral Configuration" a jako typ zařízení vyberte "Grip- per Device". Konfigurační informace chapadla jsou rozděleny na výrobce chapadla, typ chapadla, verzi softwaru a místo montáže. Specifické výrobní požadavky na konfiguraci odpovídajících informací o čelistech. Pokud uživatel potřebuje změnit konfiguraci, nejprve vyberte odpovídající číslo chapadla, kliknutím na tlačítko "Clear" (Vymazat) vymažte příslušné tlačítko a znovu proveďte konfiguraci podle potřeby;

Equipme	Gripper equip	۳	
Gripper	ROBOTIQ	•	
Gripper t	2F-85	•	
Software	R2.0	•	
Mount lo	End port 1	•	

Obrázek 5.1-1 Konfigurace čelistí chapadla

Důležité: Před kliknutím na tlačítko Clear Configuration (Vymazat konfiguraci) by měl být příslušný chapadlo neaktivní.

**Krok2Po** dokončení konfigurace chapadla si uživatel může zobrazit příslušné informace o chapadle v tabulce informací o chapadle v dolní části stránky. Pokud jsou nalezeny chyby v konfiguraci, klikněte na tlačítko "Resetovat" a znovu nakonfigurujte chapadla;

Obrázek 5.1-2 Informace o konfiguraci chapadla

**Krok3Zvolte** nakonfigurovaný chapadlo a klikněte na tlačítko "Resetovat". Po zobrazení stránky a úspěšném odeslání příkazu klikněte na tlačítko "Aktivovat" a zkontrolujte stav aktivace v tabulce s informacemi o chapadle, zda byla aktivace úspěšná;

Důležité: Když je chapadlo aktivováno, nesmí mít uchopovací předmět.

**Krok4Vyberte** příkaz "Gripper" v příkazovém rozhraní programu pro výuku. V příkazovém rozhraní chapadla může uživatel vybrat číslo chapadla, které má být ovládáno (chapadlo, které bylo nakonfigurováno a aktivováno), a nastavit odpovídající stav otevření a zavření, rychlost otevření a zavření a maximální otevírací a zavírací moment, které čekaly na pohyb chapadla. čas. Po dokončení nastavení klepněte na tlačítko Přidat aplikaci. Kromě toho lze přidat příkazy pro aktivaci a resetování chapadla, které slouží k deaktivaci/resetu chapadla za běhu programu.

Obrázek 5.1-3 Úprava příkazu Gripper

Reset		 Active
	~	
Gripper Manu.	0	
Gripper type	0	
Software	0	
version	0	
Mount location	1 26	

	G	ipper	×
Jaw No	1		
Gripper position	0	0	100
Opening and closing speed	0	0	100
Opening and closing torque	0	0	100
Maximum time	0	ms	
Whether it is blocked	block	×	
		60	Add
Jaw No		1	•
Reset			Active

## 1.3.6.1.2 Výuka programu Gripper

S/N	Formát pokynů	poznámky
1	PTP(template2,100,-1,0)	#Čekání na pinch point
2	PTP(template1,100,-1,0)	#Pinch point
3	MoveGrip- per(1,255,255,0,1000,0)	#Zavřené upínací čelisti
4	PTP(šablona2,100, -1,0)	/
5	PTP(template3,100, -1,0)	#Čekání na bod umístění
6	PTP(template3,100, -1,0)	#Bod umístění
7	MoveGrip- per(1,0,255,0,1000,0)	#Otevřené upínací čelisti

## 1.3.6.2 Obvodová konfigurace stříkací pistole

## 1.3.6.2.1 Kroky konfigurace periferie stříkací pistole

**Krok 1Vyberte** tlačítko "Konfigurace stříkací pistole" v konfiguračním rozhraní uživatelských periferií a uživatel může rychle nakonfigurovat DO potřebný pro stříkání pomocí tlačítka konfigurace funkce stříkání jedním tlačítkem (výchozí konfigurace DO10 je spuštění a zastavení stříkání a DO11 je stříkací čisticí pistole). Uživatelé si také mohou v rozhraní "IO Configuration" (Konfigurace IO) přizpůsobit DO podle vlastních potřeb;

**Důležité:** Před použitím funkce stříkání je nutné nejprve vytvořit odpovídající souřadnicový systém nástroje a při výuce programu použít vytvořený souřadnicový systém nástroje.

**Krok2Po** dokončení konfigurace klikněte na čtyři tlačítka "Start Spraying", "Stop Spraying", "Start Cleaning the Gun" a "Stop Cleaning the Gun" a vylaďte stříkací pistoli;

IO configuration	
	Set
Spray gun commissioning	
Stop spraying	Start spraying
and the second second	Chast out



**Krok3Vyberte** příkaz "spray" na příkazovém rozhraní programu pro výuku. Podle specifických požadavků na výuku programu přidejte a na odpovídající místa aplikujte čtyři příkazy "start spraying", "stop spraying", "start cleaning the gun" a "stop cleaning the gun".

Spray	×
Start spraying	Add
Stop spraying	Add
Clear gun	Add
Stop clearing	Add
Added Commands:	
	Apply

Obrázek 5.2-2 Úprava příkazů stříkací pistole

## 1.3.6.2.2 Výuka programu Spray

S/N	Formát pokynů	poznámky
1	Lin(template1,100,-1,0,0)	#Bod zahájení postřiku
2	SprayStart()	#Začít malovat
3	Lin(template2,100,-1,0,0)	#Spray path
4	Lin(template3,100,-1,0,0)	#Bod zastavení postřiku
5	SprayStop()	#Stop spraying
6	Lin(template4,100,-1,0,0)	#Bod čištění zbraně
7	PowerCleanStart()	#Začněte čistit zbraň
8	WaitTime(5000)	#Čas čištění ms
9	PowerCleanStop()	#Přestat čistit zbraň

## 1.3.6.3 Periferní konfigurace svářečky

### 1.3.6.3.1 Periferní konfigurace svářečky

Krok1V rozhraní pro konfiguraci uživatelských periferií vyberte tlačítko "konfigurace svářečky" a uživatel může rychle nakonfigurovat DI a DO požadované svářečkou pomocí tlačítka IO svářečky (výchozí konfigurace je DI12 pro signál úspěšného spuštění oblouku, DO9 pro signál přívodu plynu a DO10 pro signál přívodu plynu). Signál oblouku, DO11 běh podávání drátu, DO12 zpětné podávání drátu, DO13 volba JOB 1, DO14 volba JOB 2, DO15 volba JOB 3). Uživatelé si také mohou přizpůsobit konfiguraci v rozhraní "IO Configuration" podle vlastních potřeb;

**Důležité:** Před použitím funkce svařovacího stroje je nutné nejprve vytvořit odpovídající souřadnicový systém nástroje a při výuce programu použít vytvořený souřadnicový systém nástroje. Funkce svářečky se často používá ve spojení s laserovým sledovacím senzorem.

Krok2Po dokončení konfigurace vyberte číslo, nastavte čekací dobu a kliknutím na šest tlačítek "konec oblouku", "start oblouku", "plyn", "vypnutí plynu", "posuv drátu vpřed" a "zpětný posuv drátu" pokračujte v ladění svářečky;

Welding ma	achine confi	guration
IO configura	ition	
		Set
Welding ma	chine commi	issioning
Select No.:	0	¥
Waiting t	1000	MS
End arc		Arc
Gas OFF		Gas ON
Stop forward	1	Forward
Stop reverse		Reverse

Obrázek 5.3-1 Konfigurace svařovacího stroje

**Krok3Vyberte** příkaz "Weld" v příkazovém rozhraní programu pro výuku. Podle specifických požadavků na výuku programu přidejte a použijte na příslušných místech příkazy "začátek oblouku" a "konec oblouku".

Obrázek 5.3-2 Úprava příkazů svařovacího stroje

## 1.3.6.3.2 Výuka svářečského programu

S/N	Formát pokynů	poznámky
1	Lin(template1,100,-1,0,0)	#Výchozí bod oblouku
2	ARCStart(0,1000)	#Začít oblouk
3	Lin(template2,100,-1,0,0)	#Výchozí bod oblouku
4	ARCEnd(0,1000)	#Zastavení oblouku

Weld	[	×
Welding process No	0	•
Maximum waiting time	10000	MS
Arc	E	nd arc
Added Commands:		

### 1.3.6.4 Konfigurace periferie senzoru

### 1.3.6.4.1 Kroky konfigurace periferie senzoru

**Krok1V** rozhraní konfigurace uživatelských periferií **vyberte** tlačítko "Konfigurace senzoru". Tato část bere jako příklad konec robota. Uživatel nejprve nastaví maximální rozdíl. Maximální rozdíl bodů odchylky snímání senzoru se doporučuje nastavit standardně na hodnotu 4. Zpracování dat je založeno na skutečném použití. Na scéně se vybírají surová data nebo data YZ. IP adresa řídicí jednotky je ve výchozím nastavení 192.168.57.2, IP adresa snímače může být nakonfigurována ve stejném síťovém segmentu, port je 5020 a perioda vzorkování se doporučuje 25. Komunikační protokol je v současné době přizpůsoben komunikačnímu protokolu Ruiniu, stačí načíst odpovídající protokol. Po dokončení načítání lze senzor otestovat stisknutím tlačítek "Sensor On" (Senzor zapnut) a "Sensor Off" (Senzor vypnut).

Obrázek 5.4-1 Konfigurace IP senzoru laserového sledování

#### Důležité:

- 1. Před použitím funkce snímače je nutné nejprve vytvořit odpovídající souřadnicový systém nástroje a při výuce programu použít vytvořený souřadnicový systém nástroje. Funkce svářečky se obvykle používá se snímači.
- 2. Maximální odchylka snímače je maximální odchylka mezi polohou laserového snímání svaru v předchozím okamžiku a v aktuálním okamžiku, rozsah je [0~10], jednotka je mm a doporučená hodnota je 4.

### Krok2C Referenční bod kalibračního senzoru.

Ve funkci nastavení souřadného systému nástroje kalibrujeme nástroj typu senzor a pomocí šestibodové metody nastavujeme souřadný systém senzoru. Vybereme pevný bod v pracovním prostoru robota, posuneme středový bod senzoru do vybraného bodu M om tři různé úhly a nastavíme body 1, 2 a 3 v tomto pořadí. Přesuňte středový bod senzoru svisle přímo nad vybraný bod a zaznamenejte bod 4. Přesuňte středový bod senzoru M om pevný bod do bodu ve směru osy X souřadného systému senzoru a nastavte bod 5. Vraťte se k pevnému bodu, přesuňte se svisle nahoru a přesuňte středový bod snímače M om pevný bod do bodu ve směru osy Z souřadného systému snímače a nastavte jej jako bod 6. Klepnutím na tlačítko Vypočítat získáte polohu nástroje snímače a klepnutím na tlačítko Použít ji dokončete.

Obrázek 5.4-2 Konfigurace referenčního bodu - šestibodová metoda

**Osmibodová** metodaVe funkci nastavení souřadného systému nástroje kalibrujeme nástroj typu snímač, ke konfiguraci souřadného systému snímače použijeme osmibodovou metodu, zvolíme osmibodovou metodu, posuneme laserovou čáru snímače tak, aby se shodovala s kalibrační čárou na kalibrační desce, a udržujeme snímač a kalibrační čáru co nejblíže.

Sensor Con	figuration	
Maximu	0	Configure
Data pro	raw data 🔹	Configure
Sens <mark>itivity</mark> c	oefficient	
X-directi	NaN	
Y-direction:	NaN	
Z-directi	NaN	
		Set
Communica	tion configuration	
Communica Controll	tion configuration	Configure
Communica Controll Sensor IP:	tion configuration	Configure
Communica Controll Sensor IP: Port:	tion configuration	Configure
Communica Controll Sensor IP: Port: Samplin	tion configuration	Configure Configure
Tool Type:	sensor 🔻	
---------------------	-----------------------	-------------------
Sensor fi	Robot end 🔹	
Modify Wiza	rd	
Six point method	Eight point method	Five point method
		Setpoint 1
		Setpoint 2
		Setpoint 3
R	ТСР	Setpoint 4
4		Setpoint 5
		Setpoint 6
		Calculate

Blízká vzdálenost a kalibrační bod je rozpoznán, zaznamenejte bod 1, posuňte se asi o 20 mm ve směru -y/+y, nastavte robota tak, aby laser rozpoznal kalibrační bod, zaznamenejte bod 2, posuňte se asi o 20 mm ve směru -x/+x a nastavte robota tak, aby laser rozpoznal kalibrační bod, zaznamenejte bod 3, posuňte se asi o 20 mm ve směru -y/+y, nastavte robota tak, aby laser rozpoznal kalibrační bod, zaznamenejte bod 4, posuňte se asi o 5 mm ve směru -rx, nastavte robota tak, aby laser rozpoznal kalibrační bod, zaznamenejte bod 5. Pohybujte se asi 5 mm ve směru -ry, nastavte robota tak, aby laser rozpoznal kalibrační bod, zaznamenejte bod 6, pohybujte se asi 5 mm ve směru -rz, nastavte robota tak, aby laser rozpoznal kalibrační bod, zaznamenejte bod 6, pohybujte se asi 5 mm ve směru -rz, nastavte robota tak, aby senzor rozpoznal kalibrační bod, zaznamenejte bod 7, pohybujte se asi 5 mm ve směru -rz a nastavte robota tak, aby laser rozpoznal kalibrační bod, zaznamenejte bod 8. Kliknutím na tlačítko Calculate (Vypočítat) získáte polohu senzoru, kliknutím na tlačítko Apply (Použít) dokončete.



Obrázek 5.4-3 Konfigurace referenčního bodu - metoda osmi bodů

**Pětibodová metodaVe** funkci nastavení souřadného systému nástroje kalibrujeme nástroj typu senzor a ke konfiguraci souřadného systému senzoru použijeme pětibodovou metodu. Nejprve určíme pevný bod, zarovnáme konec nástroje s tímto bodem, zaznamenáme bod 1 a poté upravíme polohu robota tak, aby laser rozpoznal Záznam pevných bodů, respektive zaznamenáme body 2 až 5, přičemž si uvědomíme, že změna polohy musí být co největší. Klepnutím na tlačítko Calculate (Vypočítat) získáte polohu senzoru a klepnutím na tlačítko Apply (Použít) ji dokončíte.

Obrázek 5.4-4 Konfigurace referenčního bodu - metoda pěti bodů



## 1.3.6.4.2 Funkce sledování laserovým senzorem

Popis příkazu: V rozhraní příkazů pro výuku programu vyberte příkaz "Laser". Integruje pokyny týkající se laseru. Podle konkrétních požadavků na výuku programu přidejte na příslušná místa instrukce. Viz níže uvedený příklad programu.

Obrázek 5.4-5 Úprava příkazu pro sledování laserem

S/N	Formát pokynů	poznámky
1	LTLaserOn(2)	#Zapnutí senzoru
2	PTP(template1,100,-1,0)	#Výchozí bod senzoru
3	LTSearch-	#Začněte hledat
	Start(1,20,100,10000,2)	
4	LTSearchStop()	#Přestat hledat
5	Lin(seamPos,20,-1,0,0,0)	#Začátek svaru
6	LTTrackOn(2)	#Laserové sledování
7	ARCStart(0,10000)	#Svařovací oblouk svářečky
8	Lin(SeamEnd11,10-1,0,0)	#Konec svaru
9	ARCEnd(0,10000)	#Uhasínání oblouku svářeče
10	LTTrackOff()	#Sledování senzoru vypnuto
11	LTLaserOff()	#Vypněte senzor

Laser	×
Sensor command	
Select weld type	0
Open sensor	Close sensor
Sensor loading	
Function selection	Ruiniu RRT-SV2- *
Unioad	Load
Trace command	
Coordinate system name	•
Start Tracking	cease tracking
data record	
Function selection	stop recording
waiting time	10 ms
	Add

# 1.3.6.4.3 Funkce reprodukce trajektorie laserového senzoru

Popis příkazu: V příkazovém rozhraní programu pro výuku vyberte příkaz "LT-Rec". Tento příkaz slouží především k získání počátečního a koncového bodu dráhy laserového rozpoznávání a k opětovnému zobrazení trajektorie. Podle konkrétních požadavků na výuku programu přidejte instrukce na příslušná místa. Viz níže uvedený příklad programu.

LT-	Rec	×
Get weld points		
Sports mode	PTP	۲
Speed	30	%
et weld end poi	t w	eld start po
Weld data record		
Function selection	Track reap	pearar 🔹
waiting time	10	ms
Speed	30	%
		Add
Tracking recurrence		
	rac	king reproc
Added Commands:		
		Apply

Obrázek 5.4-6 Úprava příkazu pro znovuobjevení stopy

S/N	Formát pokynů	poznámky
1	PTP(template1,100,-1,0)	#Přesun do výchozího bodu
2	LaserSensor- Záznam(2,0,30)	#Senzor spustí nahrávání
3	Lin(template2,100,-1,0,0)	#Přesun na konec
4	LaserSensor- Záznam(0,0,30)	#Zastavení nahrávání
5	pos={}	#Inicializace pole
6	pos=GetWeldTrackingReco	#Výchozí bod pro získání laserového záznamu
7	If type(pos) == "table" then	#Judge datový typ
8	LaserPTP(#pos,pos)	#Přesun na počáteční bod laserové stopy
9	konec	
10	LaserSensor- Záznam(3,0,30)	#Nastavit stopu opakování
11	MoveLTR()	#Začít reprodukovat stopu
12	LaserSensor- Záznam(0,0,30)	#Konec

# 1.3.6.5 Rozšířená konfigurace periferií osy

### 1.3.6.5.1 Kroky konfigurace rozšířených periferií osy

**Krok** 1Vyberte tlačítko "Extended Axis" v rozhraní konfigurace uživatelských periferií pro vstup do rozhraní rozšířené osy, vyberte rozšířenou osu číslo 1 a kliknutím na tlačítko "Parameter Configuration" vstupte do pravého rozhraní. Nastavte typ osy, směr osy, rychlost chodu, zrychlení, limit dopředu, limit dozadu, předstih, rozlišení snímače, posunutí počátečního bodu, výrobce, model a režim a kliknutím na tlačítko Configure dokončete konfiguraci.

Obrázek 5.5-1 Rozšířená konfigurace parametrů osy

**Důležité:** Před použitím funkce rozšířené osy je nutné vytvořit odpovídající souřadnicový systém rozšířené osy a při výuce programu použít vytvořený souřadnicový systém nástroje. Funkce rozšířené osy se používá především ve spojení s funkcí svářečky a funkcí laserového sledovacího senzoru.

Krok2Kliknutím **na** tlačítko "Zero Setting" (Nastavení nuly) vstupte do vyskakovacího okna pro nastavení nuly, jak je znázorněno na obrázku vpravo. Nastavte metodu návratu nuly, rychlost hledání nuly, rychlost nulové obruče a směr osy, klikněte na tlačítko "Nastavení", prodloužená osa se začne vracet na nulu, stav návratu nuly se zobrazí v prázdném prostoru pod směrem osy, když se objeví "návrat nuly dokončen" Výzva znamená, že nulový bod prodloužené osy je úspěšně nastaven.

Obrázek 5.5-2 Rozšířené nastavení nulového bodu osy

Krok3Zvolte číslo rozšířené osy, jejíž parametry byly nakonfigurovány, klikněte na tlačítko "Servo Enable", nastavte rychlost chodu, zrychlení a maximální vzdálenost jednoho chodu a otestujte rozšířenou osu pro otáčení vpřed a vzad.

#### Obrázek 5.5-3 Rozšířená zkouška osy

Krok4Prodlužovací osa se obvykle používá ve spojení s laserovým senzorem. Laserový snímač se v tomto okamžiku obvykle instaluje externě. Konfiguraci referenčního bodu snímače je třeba kalibrovat tříbodovou metodou namísto dříve používané šestibodové metody. Vyrovnejte střed nástroje se středním bodem spodní části pravého průřezu (strana blízko kamery), nastavte bod 1, vyrovnejte střed nástroje se středním bodem spodní části druhého průřezu, což je střední bod levého průřezu, nastavte bod 2 a nastavte střed nástroje s

Shaft type	Rotation ax	is •
Axis direction	just	,
Running speed	2000	mm/s
Acceleration	1000	mm/s2
Positive direction limit	150	
Reverse direction limit	-110	
Lead	36.000	
Encoder resolution	4000	
Start offset	5	mm
manufactor	Hechuan	,
model	SV-X2EA15	50A-A
pattern	Absolute po	sition s •



Extended axis parameter configuration

Ext. axis:	1 •	
		Pick up info.
Set zero		Para. config.
Extended as	kis test	
Ext. axis:	1 •	]
Running	100	%
Accelera	100	%
Max dist	50	](mm)(°)
Disable	42 	Enable
Stop	Reverse	Forward

Přesuňte bod na prostřední bod horního okraje průřezu na pravé straně snímače, nastavte bod 3, proveďte výpočet a uložení a kliknutím na tlačítko Použít dokončete tříbodovou kalibraci.



Obrázek 5.5-4 Tříbodová kalibrace snímače

**Krok5Vyberte** příkaz "EAxis" na příkazovém rozhraní programu pro výuku. Podle konkrétních požadavků na výuku programu přidejte na příslušná místa instrukce.

	Obrázek 5.5-5	5 Rozšířená	editace	příkazů	osy
--	---------------	-------------	---------	---------	-----

#### 1.3.6.5.2 Rozšířená osa s výukovým programem pro svařování s laserovým sledováním

S/N	Formát pokynů	poznámky
1	EXT_AXIS_PTP(1,1,laserst	# Počáteční bod laserového snímače pohybu vnější osy
2	PTP(laserstart,10,-1,0)	#Výchozí bod oMobot pohybový laserový senzor
3	LTSearch-	#Začněte hledat
	Start(3,20,10,10000)	
4	LTSearchStop()	#Přestat hledat
5	EXT_AXIS_PTP(1,1,seamP	#Začátek pohybu vnější osy svaru
6	Lin(seamPos,20,-1,0,0,0)	#Startovní bod oMobot pohybující se svar
7	LTTrackOn()	#Laserové sledování
8	ARCStart(0,10000)	#Svařovací oblouk svářečky
9	EXT_AXIS_PTP(1,1,lasere	#Koncový bod sváru pohybu vnější osy
10	Lin( laserend,10,-1,0,0)	#Koncový bod oMobot pohybující se svar
11	ARCEnd(0,10000)	#Uhasínání oblouku svářeče
12	LTTrackOff	#Laserové sledování vypnuto

Mouon command		
Sports mode	asynchronous •	
Point Name:	cvrCatchPoint •	
Tool coordinate system:	toolcoord0	
Workpiece coordinate system	0	
E1	0	
E2	0	
E3	0	
E4	0	
Commissioning speed	100 %	
	Add	

#### 1.3.6.6 Konfigurace sledování dopravníku

#### 1.3.6.6.1 Kroky konfigurace sledování dopravníku

**Krok1Vyberte** tlačítko "Sledování dopravníkového pásu" v konfiguračním rozhraní uživatelských periferií pro vstup do konfiguračního rozhraní sledování dopravníkového pásu, klikněte na tlačítko "Konfigurace IO dopravníkového pásu" pro rychlou konfiguraci IO požadovaných pro funkci dopravníkového pásu a poté nakonfigurujte odpovídající parametry podle skutečného použití funkce. Zde není žádná vizuální Vezměte si jako příklad funkci sledování a uchopení, je třeba nakonfigurovat kanál snímače dopravníkového pásu, rozlišení, vedení, vizuální shodu, vybrat Ne a kliknout na tlačítko Configure.

#### Obrázek 5.6-1 Konfigurace dopravníku

**Krok2Dále** nastavte hodnotu kompenzace bodu uchopení, což je kompenzační vzdálenost ve třech směrech X, Y a Z, kterou lze nastavit podle aktuální situace během ladění.

#### Obrázek 5.6-2 Konfigurace kompenzace záchytného bodu dopravníku

**Krok3Zapněte** pásový dopravník, přesuňte kalibrovaný předmět do definovaného bodu A a zastavte pásový dopravník. Přesuňte robota, zarovnejte ostrý bod kalibrační tyče na konci robota s ostrým bodem kalibrovaného objektu, klikněte na tlačítko počátečního bodu A, zobrazí se dialogové okno se zobrazením aktuální hodnoty snímače a polohy robota a kliknutím na tlačítko Kalibrovat dokončete kalibraci počátečního bodu A.

#### Obrázek 5.6-3 Konfigurace výchozího bodu A

Krok4Kliknutím na tlačítko referenčního bodu zadejte kalibraci referenčního bodu. Při záznamu referenčního bodu zaznamenejte výšku a polohu robota při uchopení. Při každém sledování bude robot sledovat a uchopovat s oblastí výšky a polohy zaznamenaného referenčního bodu. Může se lišit M om bodu AB. Klepnutím na tlačítko Kalibrovat dokončete kalibraci referenčního bodu.

#### Obrázek 5.6-4 Konfigurace referenčního bodu

Krok5Zapněte pásový dopravník, přesuňte kalibrovaný předmět do definovaného bodu B a zastavte pásový dopravník. Přesuňte robota, vyrovnejte ostrý hrot kalibrační tyče na konci robota s ostrým hrotem předmětu, který

IO configura	ation		
			Config. IC
Function co	nfiguration		
Function	Tracking cap	oti 🔹	
Para. config	).		
Encoder	Channel 1	۲	
Resolvin	NaN		
Lead:	NaN		mm
1995 - 1996	no	Ŧ	

Grab point compensation

X::	0	mm
Y::	0	mm
Z::	ol	mm
		Configure

Sta	rtpoint a	×
x	0	
Y	0	
z	0	
RX	0	
RY	0	
RZ	0	
Encoder value	0	

	Ref. point	×
x	0	
Y	0	
z	0	
RX	0	
RY	0	
RZ	0	
	C	alibrate

klikněte na tlačítko koncového bodu B, zobrazí se dialogové okno se zobrazením aktuální hodnoty snímače a polohy robota, kliknutím na kalibraci dokončíte kalibraci koncového bodu B.

x	0	
Y	0	
z	0	
RX	0	
RY	0	
RZ	0	-
Encoder value	0	

Obrázek 5.6-5 Konfigurace terminálu B

S/N	Formát pokynů	poznámky
1	PTP(conveyorstart,30,-1,0)	#Robot chytá výchozí bod
2	While(1) do	#Loop Grab
3	Zjištění dopravníku(10000)	#IO detekce objektů v reálném čase
4	ConveyorGetTrackData(1)	#Získání polohy objektu
5	ConveyorTrackStart(1)	#Začátek sledování dopravníku
6	Lin(cvrCatchPoint,10,- 1,0,0)	#Robot dosáhne bodu úchopu
7	MoveGrip- per(1,255,255,0,10000)	#Gripper claw chytá předměty
8	Lin(cvrRaisePoint,10,- 1,0,0)	#Zvedání robotů
9	ConveyorTrackEnd()	#Konec sledování pásu
10	PTP(dopravníkzvýšení,30, - 1,0)	#Robot přijíždí na místo zadržení
11	PTP(conveyorend,30,-1,0)	#Robot dosáhne bodu umístění
12	MoveGrip- per(1,0,255,0,10000)	#Gripper release
13	PTP(conveyorstart,50,-1,0)	#Robot se opět vrátí do výchozího bodu a čeká na další zachycení.
14	konec	#Konec

# 1.3.6.6.2 Výukový program pro sledování dopravníkového pásu

# 1.3.6.7 Adaptivní konfigurace

#### 1.3.6.7.1 Kroky konfigurace adaptace polohy

**Krok 1Vyberte** tlačítko "Tracking attitude configuration" v rozhraní konfigurace uživatelských periferií pro vstup do rozhraní konfigurace nastavení polohy, vyberte typ desky a skutečný pracovní směr robota, nastavte polohu robota a nastavte bod polohy A, bod polohy B a bod polohy C, obvykle A je bod polohy roviny, B je bod polohy vzestupné hrany a C je bod polohy sestupné hrany.

Obrázek 5.7-1 Konfigurace nastavení sklonu

**Důležité:** Změna polohy mezi polohou A a polohou B, polohou A a polohou C je co nejmenší **z a p** ř **e** d **p** o **k** l **a** d **u**, ž **e** jsou splněny požadavky aplikace. Funkce přizpůsobení polohy je pomocnou funkcí aplikace, která se obvykle používá ve spojení se sledováním švu.

**Krok2Vyberte** příkaz "Adjust" v příkazovém rozhraní programu pro výuku. Podle konkrétních požadavků na výuku programu přidejte pokyny na příslušná místa.

Obrázek 5.7-2 Úprava příkazu pro nastavení polohy

Sheet ty	Corrugated pl •	
Direction	Left to right	
		Point a
		Point b
	[	Point c
•	<b>_</b>	
	1110	

# 1.3.6.7.2 Samočinná adaptace s rozšířenou osou a výukovým programem pro svařování s laserovým sledováním

S/N	Formát pokynů	poznámky
1	EXT_AXIS_PTP(1,1,laserst	# Počáteční bod laserového snímače pohybu vnější osy
2	PTP(laserstart,10,-1,0)	#Výchozí bod oMobot pohybový laserový senzor
3	LTSearch- Start(3,20,10,10000)	#Začněte hledat
4	LTSearchStop()	#Přestat hledat
5	EXT_AXIS_PTP(1,1,seamP	#Začátek pohybu vnější osy svaru
6	Lin(seamPos,20,-1,0,0,0)	#Startovní bod oMobot pohybující se svar
7	LTTrackOn()	#Laserové sledování
8	ARCStart(0,10000)	#Svařovací oblouk svářečky
9	PostureAdjus- tOn(0,PosA,PosC,PosB,100	#Attitude adaptive adjustment on
10	EXT_AXIS_PTP(1,1,lasere	#Koncový bod sváru pohybu vnější osy
11	Lin( laserend,10,-1,0,0)	#Koncový bod oMobot pohybující se svar
12	ARCEnd(0,10000)	#Uhasínání oblouku svářeče
13	PostureAdjustOff(0)	#Attitude adaptive adjustment off
14	LTTrackOff	#Laserové sledování vypnuto



## 1.3.6.8 Konfigurace periferie snímače síly a točivého momentu

#### 1.3.6.8.1 Kroky konfigurace snímače síly a točivého momentu

Krok1V rozhraní konfigurace uživatelských periferií **vyberte** tlačítko "End Peripheral Configuration" a jako typ zařízení vyberte "Force Sensor Device". Informace o konfiguraci snímače síly jsou rozděleny na výrobce, typ, verzi softwaru a místo montáže. Nakonfigurujte odpovídající informace o snímači síly. Pokud uživatel potřebuje konfiguraci změnit, může nejprve vybrat odpovídající číslo, kliknutím na tlačítko "Clear" vymazat odpovídající informace a znovu provést konfiguraci podle potřeby;

Equipme	Force sensor	۲
manufac	kunwei	۲
type:	KWR75B	Ŧ
Software	1.0	T
Mount lo	End port 1	¥

Obrázek 5.8-1 Konfigurace snímače síly/motoru

<b>)ůležité:</b> Před kliknutím na tlačítko	Vymazat konfiguraci b	y měl být příslušny	ý senzor neaktivní.
---	-----------------------	---------------------	---------------------

**Krok2Po** dokončení konfigurace snímače síly si uživatel může zobrazit odpovídající in- formaci snímače síly v informační tabulce v dolní části stránky. Pokud je nalezena chyba v konfiguraci, může uživatel kliknout na tlačítko "Resetovat" a provést novou konfiguraci.

#### Obrázek 5.8-2 Informace o konfiguraci snímače síly/motoru

Krok3Zvolte číslo nakonfigurovaného snímače síly a klikněte na tlačítko "Resetovat". Po zobrazení stránky a úspěšném odeslání příkazu klikněte na tlačítko "Aktivovat" a zkontrolujte stav aktivace v tabulce informací o snímači síly, abyste zjistili, zda je aktivace úspěšná; kromě toho se snímač síly Tam je počáteční hodnota a uživatel může vybrat "korekci nulového bodu" a "odstranění nulového bodu" podle požadavků na použití. Při korekci nulového bodu snímače síly je třeba zajistit, aby byl snímač síly ve vodorovné a svislé poloze a aby robot nebyl vybaven zátěží.

Krok4Po dokončení konfigurace snímače síly je třeba nakonfigurovat souřadnicový systém nástroje typu snímače a hodnotu souřadnicového systému nástroje snímače lze přímo zadat a použít podle vzdálenosti mezi snímačem a středem koncového nástroje.

orce sensor	guied	1 •
Reset		Active
Remove zero		Zero correctio
manufacturer	0	
type	0	
Software	0	
version	0	
Mount		

#### 1.3.6.8.2 Identifikace zatížení snímače síly a točivého momentu

Rozpoznání specifické polohy: vymažte data koncového zatížení, nakonfigurujte snímač síly, nastavte souřadnicový systém snímače, upravte koncovou polohu robota tak, aby byla svisle dolů, proveďte "korekci nulového bodu" a nainstalujte koncové zatížení. Nejprve zvolte odpovídající souřadnicový systém snímače a nástroje, nastavte robota tak, aby snímač a nástroj byly svisle dolů, zaznamenejte data a vypočítejte kvalitu. Poté nastavte tři různé polohy robota, zaznamenejte tři sady dat v tomto pořadí, vypočítejte střed hmotnosti a po potvrzení správnosti klikněte na tlačítko Použít.

**Dynamická identifikacePo** vymazání dat koncového zatížení a konfiguraci snímače síly nastavte souřadnicový systém snímače, upravte koncovou polohu robota tak, aby byla svisle dolů, proveďte "korekci nulového bodu" a nainstalujte koncové zatížení. Klikněte na tlačítko "Identification On", přetáhněte robota k pohybu a poté klikněte na tlačítko "Identification Of", výsledek zatížení může být automaticky aplikován na robota.

Obrázek 5.8-4 Identifikace zatížení snímače síly/motoru

#### 1.3.6.8.3 Snímač síly a točivého momentu s asistencí při přetahování

Po konfiguraci senzoru jej lze použít spolu se senzorem pro lepší asistenci tažnému robotu. Při prvním použití jej můžete nakonfigurovat podle údajů na obrázku vpravo. Po dokončení aplikace nemusíte v tuto chvíli vstupovat do režimu tažení a můžete přímo přetahovat senzor koncové síly a ovládat robota tak, aby se pohyboval v pevné poloze.

Obrázek 5.8-4 Snímač síly a točivého momentu s aretací tahu

**Důležité:** Koeficient nastavení je koeficient měkkosti a jeho rozsah je [0~3]. Čím větší je hodnota, tím detailnější je efekt měkkosti. Doporučuje se nastavit přiměřený koeficient podle skutečné situace a obecně doporučená hodnota je 1.

Automatic load identification of force/torque
sensor

Sensor	•	
Mass measurem data of sensors a	ent, recording vertic and tools	cal downward
Record data		Calculate
Load weight		kg
Center of mass i	neasurement, recor	ding 3 groups of
data under differ	ent attitudes	
Record data 1	Record data 2	Record data 3
X	Y	Ζ
Calculate		Apply
Automatic load	d identification of	force/torque
CONDOL:		
Identify OFF		Identify ON

sampling period 0

ms

Set

Drag	teachi	ng lock	configu	ration	(
Drag I	lock				
Paran	neter s	etting o	f each de	gree o	of freedom
X	0.005	Y	0.005	Z	0.005
RX	)	RY	0	RZ	0
Drag t switch	the tea	ching fu	unction	Enat	Apply ble
Drag f	the tea າ	ching fu	inction	Enat	Apply ble •
Drag I switch Force	the tea 1 senso	ching fu	unction dary lock	Enat	Apply ble •
Drag 1 switch Force Adjust	the teanso	ching fur r secon	unction dary lock	Enat	Apply ble •
Drag I switch Force Adjust	the tea senso tm	ching fu r secon 1 open	unction dary lock	Enat ing	Apply

## 1.3.6.8.4 Snímač síly/točivého momentu Detekce kolize

Popis příkazu: Příkaz "FT\_Guard" je příkaz pro detekci kolizí. Zvolte příslušný souřadnicový systém snímače, zkontrolujte efektivní detekci směru točivého momentu, nastavte aktuální hodnotu, maximální práh kolize a minimální práh kolize. Normální rozsah podmínky detekce kolize je (aktuální hodnota-minimální práh, aktuální hodnota+maximální práh), nastavené příkazy "Otevřít" a "Zavřít" jsou přidány do programu.

Co	ordin	ate system name		1
		Current value	Maximum threshold	Minimum threshold
	Fx	0.000	0	0
	Fy	0.000	0	0
	Fz	0.000	0	0
	тх	0.000	0	0
	ту	0.000	0	0
-	Tz	0.000	0	0

# Obrázek 5.8-5 Úprava příkazu FT\_Guard

S/N	Formát pokynů	i poznámky
1	FT_Guard(1,1,	#Detekce kolize silou/momentem zapnuta
2	PTP(šablona1, 1,0)	#Příkaz pohybu
3	FT_Guard(0,1,	#Detekce kolize síla/moment vypnuta

## 1.3.6.8.5 Snímač síly/točivého momentu Řízení síly Pohyb

Popis pokynů: Instrukce "FT\_Control" je instrukce pro řízení pohybu síly, která může robota přimět k pohybu v blízkosti nastavené síly a často se používá v brusných scénách. Zvolte odpovídající souřadnicový systém snímače, zkontrolujte efektivní detekci směru točivého momentu, nastavte práh detekce a proporcionální koeficient PID v každém směru (obecně nastavte p na 0,001), nastavte maximální vzdálenost nastavení (odpovídající X, Y, Z) a maximální úhel nastavení (odpovídající RX, RY, RZ), přidejte do programu instrukce "otevřít" a "zavřít".

	F/T			×
Coordinate system	name	1		•
<b>Fx</b> 0		Тх	0	
<b>Fy</b> 0		Ту	0	
<b>Fz</b> 0		Tz	0	
F_P_gain 0.0005	T_	P_gair	0	3
F_I_gain 0	T	l_gain	0	
F_D_gain_0	T	D_gaiı	<b>n</b> 0	]
Adaptive start sto stati	stop		•	
ILC control sta stop stati	art stop		T	
Maximu adjustment distan	m 0 ce		ſ	nm
Maximu adjustment ang	m 0 le		°	
close				open
			I	Reset



S/N	Formát pokynů	poznámky
1	FT_Control(1,11,1,0,1,0,0,0	#Řízení pohybu silou/kroutícím momentem zapnuto
2	Lin(template3,100,-1,0,0)	#Příkaz pohybu
3	FT_Control(0,11,1,0,1,0,0,0	#Řízení pohybu síly/kroutícího momentu vypnuto

# 1.3.6.8.6 Zavedení šroubu snímače síly/točivého momentu

Popis pokynů: Pokyn "FT\_Spiral" je průzkum a vkládání spirálového vedení, které se obecně používá pro montážní akci hřídelového otvoru válcového hřídele. Před spuštěním akce je třeba přetáhnout konec robota do přibližné polohy otvoru. Podle aktuální scény nastavte parametry příkazu a přidejte jej do programu. Po spuštění bude robot zkoumat spirálovitým pohybem.

recommended
recommended
mm
N/Nm
ms
mm/s
lelical insertio

# Obrázek 5.8-7 Úprava příkazu FT\_Spiral

S/N	Formát pokynů	poznámky
1	FT_Control(1,10,0,0,1,0,0,0	#Řízení pohybu silou/kroutícím momentem zapnuto
2	FT_SpiralSearch(0,0.7,0,60	#Vložení spirály
3	FT_Control(0,10,0,0,1,0,0,0	#Řízení pohybu síly/kroutícího momentu vypnuto

## 1.3.6.8.7 Snímač síly/motoru Otočné vkládání

Popis pokynů: Pokyn "FT\_Rot" je rotační průzkumné vložení, které se obecně používá pro šroubovicové vložení a používá se pro montáž otvoru hřídele klíče. Před spuštěním akce je třeba přesunout konec robota do otvoru nalezeného šroubovým průzkumem nebo do plně zarovnaného výukového otvoru. Podle aktuální scény nastavte parametry příkazu a přidejte jej do programu. Po spuštění se robot pomalu roztočí k průzkumu.

F/T		×
Coordinate system name	Tool coordin 🔻	
The following are advanced param values are in brackets	eters, and the recomm	nended
Rotating angular velocity	0	°/s
Trigger force or termination torque (0- 100)	50	N/Nm
Maximum rotation angle (5)	5	•
Direction of force	Direction fz 🔻	
Maximum rotation angular acceleration	0	°/s^2
Insertion direction	just 🔻	
Return	Rot	late Insert
Added Commands:		

# Obrázek 5.8-8 Úprava příkazu FT\_Rot

S/N	Formát pokynů	poznámky
1	FT_Control(1,10,0,0,1,0,0,0	#Řízení pohybu silou/kroutícím momentem zapnuto
2	FT_RotInsertion(0,3,0,5,1,0)	#Rotate Insert
3	FT_Control(0,10,0,0,1,0,0,0	#Řízení pohybu silou/kroutícím momentem vypnuto

# 1.3.6.8.8 Snímač síly/motoru Přímé vložení

Popis pokynů: Pokyn "FT\_Lin" je rotační průzkumné vložení, které se obecně používá k provádění šroubovitého vložení nebo rotačního vložení a používá se pro montáž otvoru v hřídeli klíčové hřídele. Před spuštěním akce je třeba přesunout konec robota k otvoru nalezenému šroubovým průzkumem, otočit konec akce vkládání nebo plně zarovnaný výukový otvor, nastavit parametry příkazu podle aktuální scény, přidat jej do programu a spustit Poté se robot bude pohybovat po přímce v nastaveném směru.

БЛ		×
Coordinate system name	Tool coordin •	]
The following are advanced paran values are in brackets	neters, and the recom	mended
Action termination force threshold (0-100)	50	N
Linear speed (1)	1	mm/s
Linear acceleration	0	mm/s^2
Maximum insertion distance	0	mm
Insertion direction	just 🔹	
Return	Lir	ne Insertion
Added Commands.		
		Apply
	8	

Obrázek 5.8-9 Úprava příkazu FT\_Lin

S/N	Formát pokynů	poznámky
1	FT_Control(1,10,0,0,1,0,0,0	#Řízení pohybu silou/kroutícím momentem zapnuto
2	FT_LinInsertion(0,50,1,0,1)	#Vkládání řádků
3	FT_Control(0,10,0,0,1,0,0,0	#Řízení pohybu silou/kroutícím momentem vypnuto

### 1.3.6.8.9 Snímač síly/motoru Orientace povrchu

Popis pokynů: Instrukce "FT\_FindSurface" slouží k určení polohy povrchu a obecně se používá k nalezení povrchu objektu. Podle aktuální scény nastavte odpovídající souřadnicový systém, směr pohybu, osu pohybu, lineární rychlost průzkumu, lineární zrychlení průzkumu, maximální vzdálenost průzkumu, práh síly ukončení akce a další parametry, přidejte je do programu, spusťte program, akce se začne provádět a konec robota se začne zpomalovat Pohybujte se ve směru povrchu.

F/T			×
Coordinate system name	Tool coor	din 🔻	
The following are advanced paran values are in brackets	neters, and the	recomm	ended
Moving direction	just	Y	
Move Axis	X	T	
Explore linear speed	1		mm/s
Explore acceleration	0		mm/s^2
Maximum exploration distance	0		mm
Action termination force threshold	50		Ν
Return		l	ocate
Added Commands:			
			Apply



S/N	Formát pokynů	poznámky
1	PTP(1,30,-1,0)	#Počáteční pozice
2	FT_FindSurface(0,1,3,1,0,1)	#Umístění letadla

# 1.3.6.8.10 Centrování snímače síly/motoru

Popis pokynů: Instrukce "FT\_CalCenter" slouží k určení středu a obecně se používá k nalezení středové roviny dvou ploch. Podle aktuální scény nastavte odpovídající souřadnicový systém, směr pohybu, osu pohybu, lineární rychlost průzkumu, lineární zrychlení průzkumu, maximální vzdálenost průzkumu, práh síly ukončení akce a další parametry, najděte rovinu A, respektive rovinu B, přidejte je do programu a spusťte program. Akce se začne provádět a robot se pomalu pohybuje směrem k ploše A. Po umístění na ploše A se robot pomalu pohybuje směrem k ploše B. Po umístění na ploše B lze vypočítat polohu středové roviny.

F/T	×
/id plane positioning calculation	
Start	Finish
	Return
Added Commands:	
	Apply

Obrázek 5.8-11 Úprava příkazu FT\_CalCenter

S Formát pokynů	poznámky
1 PTP(1,30,-1,0)	#Počáteční pozice
2 FT_CalCenterStart()	#Začátek polohování povrchu
3 FT_Control(1,10,0,0,1,0,0,0,0, 10,0,0,0,00001,0,0,0,0	#Řízení pohybu silou/kroutícím momentem zapnuto
4 FT_FindSurface(1,2,2,10,0,20	#Polohovací rovina A
5 FT_Control(0,10,0,0,1,0,0,0,0, 10,0,0,0,00001,0,0,0,0	#Řízení pohybu silou/kroutícím momentem vypnuto
6 PTP (1,30, - 1,0) - počáteční pozice	#Počáteční pozice
7 FT_Control(1,10,0,0,1,0,0,0,0, 10,0,0,0,00001,0,0,0,0	#Řízení pohybu silou/kroutícím momentem zapnuto
8 FT_FindSurface(1,1,2,20,0,20	#Polohovací rovina B
9 FT_Control(0,10,0,0,1,0,0,0,0, 10,0,0,0,00001,0,0,0,0	#Řízení pohybu síly/kroutícího momentu vypnuto
1 pos = {}	#Získání kartézské polohy polohovacího středu
1 pos = FT_CalCenterEnd()	#Získání kartézské polohy polohovacího středu
1 MoveCart(pos,GetActualTCPN 1,0)	#Přesun do středu polohování

#### 1.3.6.8.11 Snímač síly/točivého momentu Detekce síly poklepu

Popis pokynů: Příkaz "FT\_Click" je detekce síly kliknutí, která se používá k detekci síly kliknutí a obvykle se používá ve spojení s akcí polohování povrchu. Po nastavení parametrů jej přidejte do programu, spusťte program a konec se začne pohybovat směrem k cíli podél směru Z souřadného systému nástroje. Jakmile síla v kladném směru Z dosáhne hodnoty síly kliknutí, je detekce síly kliknutí dokončena.

F/T		×
Action termination force threshold	-	N
Linear velocity	_	mm/s
Linear acceleration		mm/s^2
Maximum insertion distance		mm
Return		Detection
Added Commands:		
		Apply

Obrázek 5.8-12 Příkaz FT\_Click Upravit

Příklad programu:

S/N	Formát pokynů	poznámky
1	PTP(1,30,-1,0)	#Počáteční pozice
2	FT_Click(0,5,5,0,100,0)	#Detekce síly v místě

#### 1.3.6.9 Rozšířená konfigurace periferních zařízení IO

#### 1.3.6.9.1 Rozšířené kroky konfigurace zařízení IO

**Krok1V** rozhraní konfigurace uživatelských periferií vyberte tlačítko "End Peripheral Configuration" a jako typ zařízení vyberte "Ex- tended IO Device". Konfigurační informace rozšířeného IO zařízení jsou rozděleny na výrobce, typ, verzi softwaru a místo montáže. Uživatelé si mohou vybrat podle konkrétních výrobních potřeb. Pro konfiguraci odpovídajících informací o zařízení. Pokud uživatel potřebuje změnit konfiguraci, může nejprve vybrat příslušné číslo, kliknutím na tlačítko "Clear" (Vymazat) vymazat příslušné informace a znovu je nakonfigurovat podle potřeby;

#### Obrázek 5.9-1 Rozšířená konfigurace zařízení IO

Důležité: Před kliknutím na tlačítko Vymazat konfiguraci by příslušné zařízení mělo být neaktivní.

Equipme	Extended IO	( 🔻
manufac	NSR	•
type:	SmartTool	•
Software	F1.0	•
Mount Io	End port 1	•

Krok2Po dokončení konfigurace rozšířeného zařízení IO může uživatel kliknutím na nabídku funkcí "Smart Tool" v pomocné aplikaci vstoupit na stránku konfigurace funkcí a uživatel může přizpůsobit funkce jednotlivých tlačítek na koncové rukojeti, včetně (nový program, podržet program, PTP , Lin, ARC, začátek tkaní, konec tkaní, port IO).

Obrázek 5.9-2 Konfigurace funkcí zařízení Extended IO

# 1.3.6.10 Konfigurace paletizačního systému

## 1.3.6.10.1 Kroky konfigurace paletizačního systému

**Krok1V** rozhraní konfigurace uživatelských periferií **vyberte** tlačítko "Konfigurace paletizačního systému". Při prvním použití je třeba nejprve vytvořit předpis. Klikněte na tlačítko "Create Recipe" (Vytvořit recepturu), zadejte název receptury, klikněte na tlačítko "Create" (Vytvořit) a po úspěšném vytvoření klikněte na tlačítko "Start Configuration" (Spustit konfiguraci). Vstupte na stránku konfigurace paletizace.

Obrázek 5.10-1 Konfigurace receptury paletizace

**Krok2** Kliknutím na tlačítko "Configure" (Konfigurace) na panelu konfigurace obrobku vstupte do vyskakovacího okna konfigurace obrobku, nastavte "délku", "šířku", "výšku" obrobku a bod uchopení obrobku, kliknutím na tlačítko "confirm configuration" (Potvrdit konfiguraci) dokončete nastavení informací o obrobku.

Obrázek 5.10-2 Konfigurace obrobků při paletizaci

Krok3Kliknutím **na tlačítko** "Configure" na panelu konfigurace zásobníku vstupte do vyskakovacího okna konfigurace zásobníku, nastavte zásobník "Mont", "side" a "height", poté nastavte stanici a přechodový bod stanice, kliknutím na tlačítko "confirm configuration" dokončete nastavení informací o zásobníku.

Obrázek 5.10-3 Konfigurace palet pro paletizaci

**Krok4Kliknutím na** "Configure" v liště konfigurace režimu vstoupíte do vyskakovacího okna konfigurace režimu, nastavíte interval obrobku a vpravo M ame je metoda umístění simulace obrobku, kterou lze přidávat jednotlivě nebo v dávkách. Poté nastavte počet vrstev paletizace a režim každé vrstvy, kliknutím na tlačítko "potvrdit konfiguraci" dokončete nastavení informací o režimu.

Obrázek 5.10-4 Konfigurace režimu paletizace

Krok4Kliknutím na "Generate Program" otevřete "Palletizing Monitoring Page", kde můžete zobrazit a prohlížet "Generation Information", "Alarm Information" a "Palletizing Program".

Obrázek 5.10-5 Monitorování paletizačního systému

en e				
Func	tion co	nfig		
A	key:	PTP	¥	
spee	d:	30	1	%
				Configure
В	key:	LIN	Y	]
spee	d:	30	1	%
				Configure
С	key:	ARC	¥	]
spee	d:	20	1	]%
				Configure
D	key:	New program	¥	
				Configure

### Palletizing system configuration

Before starting the configuration, verify that the station installation matches:









palletizing system monitoring		3/23/2023, 10:47:16 AM (1/O)
Production i	nfo.	Palletizing program
DO Signal C C00 C C C DI Signal C CI0 Production times 0 Production time 0 Single production time: 0	Curre progra Curre index Curre	rent jram. • rent layer ex: 0 ex: 0

# 1.3.7 Příloha

# 1.3.7.1 Dodatek 1: Chyby regulátoru pohybu a způsoby jejich řešení

Klasifikace chyb	Název chyby (zobrazení výukového přívěsku)
Chyba příkazového bodu	Chyba společného příkazového bodu Chyba přímky cílového bodu
	Chyba středního bodu oblouku
	Chyba cílového bodu oblouku
	Rozteč příkazových bodů oblouku je příliš malá
	Chyba ve středním bodě 1 celé kružnice/spirály (včetně odchylky nástroje) Chyba ve středním bodě 2 celé kružnice/šroubovice (včetně odchylky nástroje)
	Chyba ve středním bodě 3 celé kružnice/spirály (včetně odchylky nástroje)
	Rozteč příkazových bodů celé kružnice/šroubovice je příliš malá.
	Chyba příkazového bodu TPD
	Příkazový nástroj TPD neodpovídá aktuálnímu nástroji
	TPD Odchylka mezi aktuálním pokynem a počátečním bodem následujícího pokynu.
	Chyba přepínání interního/externího nástroje
	Překročení společného příkazu PTP
	Překročení společného velitelství TPD
	LIN ARC vydala společný příkaz k překročení
	JOG společné překročení velení
	Překročení příkazové rychlosti v kartézském prostoru
	Překročení krouticího momentu v prostoru kloubu

	Klasifikace chybNázev chyby(teaching pendant display)
	Rychlost příkazů v prostoru kloubů osv 1 - osv 6 překračuje mezní hodpotu
	Konfigurace kloubu dalšího příkazu se mění (existuií singulární pozice
	Společná konfigurace aktuálního příkazu se změnila (je zde singulární po
	Chyba instrukce, mezi ARCSTART jsou povoleny pouze instrukce LIN a ARC.
	Chyba instrukce, mezi WEAVESTART a WE je povolena pouze instrukce LIN
	Chyba parametru svařování Swing
	Odchylka povelu laserového senzoru je příliš velká
	Příkaz laserového senzoru je přerušen a sledování švu je předem ukončeno.
	Překročení rychlosti příkazu externí osy
	Nadměrná odchylka mezi vnějším příkazem osy a zpětnou vazbou
	Sledování pásu - nadměrná změna polohy mezi výchozím a referenčním bodem
	Řízení konstantní síly - směr (X, Y, Z, RX, RY, RZ) překračuje maximální
	připustnou hodnotu.
Selhani pohonu	Porucha ovladace 1-6 os
Mimo mekky limit Závada při kolizi	1-6 osa mimo mekky limit poručna Závrada koliza 1.6 páprav
Zavaua pri Kolizi	Zavada Kolize 1-6 liapiav
Cnyba	spalna verze konigurachino souboru
souboru	zul Kolliguidelli Soudol zul Se
	nepoudino nacisi chyba verze
	uzivalejskeno promu Nesprávná verze konfiguračního souboru
	evavis Nepodařilo se pačíst konfigurační
	souhor evavis
Chuha	Model robota je nekonzistentní a je třeba jej resetovat
	Chyha kanálu
10	Chyba hodnoty
	WaitDI wait timeout
	WaitAI waiting timeout
	WaitToolDI wait timeout
	WaitToolAI wait timeout
	Chyba funkce nakonfigurované
	pro kanál Časový limit pro
	stávkující oblouk
	Časový limit pro zhasnutí oblouku
	Casový limit pro vyhledání
	polohy
Chyba	Casovy limit detekce 10 dopravního pasu, resetovatelny
čelistí	Ospesny naraz obiouku Di neni nakoniigurovan
	chyba časoveno mninu nehybu čeljetí 495 timeout
	Chyba formátu příkazu
	Znoždění nohvhu musí být aktivováno před pohvhem Při
	nohybu musí být aktivní hit Tenlota příliš vysoká
	Příliš nízké nanětí
	Automatické uvolnění
	Vnitřní porucha
	Aktivace selhala
	Nadproud
	Automatické uvolnění konce
Varován	Změny v konfiguraci ramenního kloubu
<u></u>	Změny v konfiguraci loketního kloubu
1	Změny v konfiguraci zápěstního kloubu
Inicializa ce RPY se nezdařila

	Klasifikace chybNázev chyby(teaching pendant display)
Nesprávný počet aktivních podřízených stanic Chyba podřízeného zařízení	Po 1 minutě přepněte do obecného výukového režimu. Špatný počet aktivních podřízených stanic Drop line M om station Stav podřízeného zařízení neodpovídá nastavené hodnotě Slave není nakonfigurovánChybná konfigurace Slave Chyba micializace Romunikace podřízené poštovní schránky
Unozornění na beznečnostní dveře	Snuštění heznečnostních dveří
Varování před pohybem	Příkaz LIN příliš mění polohu
Varování před rušivou zónou	Vstup do rušené zóny
Chyba parametru	Chyba překročení čísla nástroje
	Chyba prahové hodnoty pro dokončení polohování
	Chyba úrovně havárie
	Chyba hmotnosti nákladu
	Chyba centroidu zatížení (X, Y, Z)
	Casová chyba filtru DI
	Casová chyba filtru AxleDl
	Chyba času filtrování Al
	Casova chyba filtrovani AxleAl
	Chyba rozsahu vysoke a nizke urovne Di
	Chyba rózsanu vysoke a nizke niadiny DO
	Chyba překročení čícla externí osv
	Sledování donravníku - chyba kanálu snímače
	Sledování dopravníku - chyba kalala osvobrohku
Metoda osmi bodů	Změna postoje u první skupiny údajů je příliš velká
	Chyba dat. výpočet se nezdařil
Porucha vnější osy mimo měkký limit	Externí osa 1-4 mimo měkkou mezní poruchu
Výuka komunikace s přívěskem (raspberry pi) selhala	Komunikace mezi webovou aplikací a přívěskem pro výuku (Raspberry Pie)
Chyba původu	Původ se změnil a je třeba jej obnovit

### 1.3.7.2 Příloha 2: Tabulka chybových kódů ovladače serva

Kód závady	Název poruchy	Způsob zpracování
1	Softwarová nadproudová porucha	<ol> <li>Zkontrolujte, zda se zatížení nebo odpor kloubu nezvětšuje nebo není abnormální;</li> <li>Pokud se závada neodstraní, opravte nebo vyměňte desku pohonu.</li> </ol>
2	Přepěťová porucha	Snížení rychlosti nebo zrychlení robota.
		pokračování na další straně

Kód závady	Název poruchy	Způsob zpracování
3	Porucha pod napětím	<ol> <li>Zkontrolujte, zda není výstupní napětí 48 V řídicí jednotky abnormální;</li> <li>Zkontrolujte, zda nedošlo ke zkratu hnací desky a kloubového pouzdra;</li> <li>Pokud se závada neodstraní, opravte nebo vyměňte desku pohonu.</li> </ol>
4	Porucha přehřátí	Snížit zatížení nebo rychlost robot.
5	Porucha při přetížení	Snižit zatižení nebo rychlost robot.
6	Porucha nadměrné rychlosti	<ol> <li>Zkontrolujte, zda není magnetické opletení a upevňovací šroub hřídele motoru uvolněný;</li> <li>Znovu proveďte kalibraci nuly snímače;</li> <li>Pokud se závada neodstraní, opravte nebo vyměňte sestavu magnetického editoru.</li> </ol>
7	Porucha abnormálního parametru Domucha při útěly	Opravte nebo vyměňte hnací desku.
o	Por ucha pri uleku	<ol> <li>Zkontrolujte, zda není magnetické opletení a upevňovací šroub hřídele motoru uvolněný;</li> <li>Znovu proveďte kalibraci nuly snímače;</li> <li>Pokud se závada neodstraní, opravte nebo vyměňte sestavu magnetického editoru.</li> </ol>
9	Chyba polohy	<ol> <li>Zkontrolujte, zda se zatížení nebo odpor kloubu nezvětšuje nebo není abnormální;</li> <li>Pokud se závada neodstraní, opravte nebo vyměňte desku pohonu.</li> </ol>
10	Porucha přetečení polohy	1. Zkontrolujte, zda je pevný limit uvolněný; 2. Znovu proveďte kalibraci nuly robota.
11	Hardwarová nadproudová porucha	Opravte nebo vyměňte hnací desku.
12	Porucha inhibice pohonu	Robot NENÍ povolen.
		pokračování na další straně

### Tabulka 2 - pokračování Máma předchozí strana

Tabulka 2 - pokračováníMáma předchozí strana				
Kód závady	Název poruchy	Způsob zpracování		
13	Porucha zablokovaného rotoru motoru	<ol> <li>Zkontrolujte, zda je brzdový elektromagnet sepnutý;</li> <li>Zkontrolujte, zda je překročen pevný limit;</li> <li>Pokud se závada neodstraní, opravte nebo vyměňte desku pohonu.</li> </ol>		
14	Výpadek napájení	Robot NENÍ povolen.		
15	Porucha STO	Robot NENI povolen.		
16	Porucha nastavení nuly fázového proudu AD	Opravte nebo vyměňte hnací desku.		
17	Porucha EEPROM	Opravte nebo vyměňte hnací desku.		
18	Porucha haly	<ol> <li>I. Zkontrolujte, zda je kabelový svazek pevně zasunut a zda nedošlo ke zkratu nebo rozpojení;</li> <li>Pokud se závada přesto neodstraní, opravte nebo vyměňte kloub.</li> </ol>		
19	Kodér selhal	Oprava nebo výměna magnetického opletení montáž.		
20	Porucha nastavení nuly snímače	<ol> <li>I. Znovu proveďte kalibraci nuly snímače;</li> <li>Pokud se závada neodstraní, opravte nebo vyměňte sestavu magnetického editoru.</li> </ol>		
21	Porucha snímače Z-fáze ztráty signálu	Robot NENÍ povolen.		
22	Porucha počítání snímače	Robot NENÍ povolen.		
23	Přetečení dat více otáček kodéru závada	Robot NENÍ povolen.		
24	Porucha externích hodin	Opravte nebo vyměňte hnací desku.		
25	Porucha sledu fází UVW	Robot NENÍ povolen.		
26	Porucha FPGA	Robot NENÍ povolen.		
27	Porucha nulového návratu	Robot NENÍ povolen.		
28	Porucha magnetického snímače	<ol> <li>Zkontrolujte, zda není magnetické opletení a upevňovací šroub hřídele motoru uvolněný;</li> <li>Pokud se závada neodstraní, opravte nebo vyměňte sestavu magnetického editoru.</li> </ol>		
		pokračování na další straně		

Kód závady	Název poruchy	Způsob zpracování
29	Porucha odpojení napájecího vedení motoru	<ol> <li>Zkontrolujte, zda je napájecí vedení motoru pevně zasunuto a zda nedošlo ke zkratu nebo rozpojení;</li> <li>Pokud se závada neodstraní, opravte nebo vyměňte desku pohonu.</li> </ol>
30	Porucha EtherCAT	<ol> <li>Zkontrolujte, zda je síťový kabel pevně zapojen a zda nedošlo ke zkratu nebo rozpojení;</li> <li>Pokud se závada neodstraní, opravte nebo vyměňte desku pohonu.</li> </ol>
31	Porucha EtherCAT_SM_DOG	<ol> <li>Zkontrolujte, zda je síťový kabel pevně zapojen a zda nedošlo ke zkratu nebo rozpojení;</li> <li>Pokud se závada neodstraní, opravte nebo vyměňte desku pohonu.</li> </ol>
32	Selhání EtherCAT_FATALSYNC	<ol> <li>Zkontrolujte, zda je síťový kabel pevně zapojen a zda nedošlo ke zkratu nebo rozpojení;</li> <li>Pokud se závada neodstraní, opravte nebo vyměňte desku pohonu.</li> </ol>
33	Porucha EtherCAT_SYNC	<ol> <li>Zkontrolujte, zda je síťový kabel pevně zapojen a zda nedošlo ke zkratu nebo rozpojení;</li> <li>Pokud se závada neodstraní, opravte nebo vyměňte desku pohonu.</li> </ol>
34	Porucha EtherCAT_RFT	<ol> <li>Zkontrolujte, zda je síťový kabel pevně zapojen a zda nedošlo ke zkratu nebo rozpojení;</li> <li>Pokud se závada neodstraní, opravte nebo vyměňte desku pohonu.</li> </ol>
35	Porucha adresy hnacího hřídele	<ol> <li>Překonfigurujte adresu osy pohonu;</li> <li>Pokud se závada neodstraní, opravte nebo vyměňte desku pohonu.</li> </ol>

#### Tabulka 2 - pokračování Máma předchozí strana

Kód závady	Název poruchy	Způsob zpracování
36	Porucha kalibrace nuly robota	<ol> <li>I. Znovu proveďte kalibraci nuly robota;</li> <li>Nejprve pomocí JLINK vymažte FLASH, poté znovu stáhněte program a vynulujte jej;</li> <li>Pokud se závada neodstraní, opravte nebo vyměňte desku pohonu.</li> </ol>
37	Selhání komunikace kodéru	<ol> <li>Zkontrolujte, zda je svazek snímače pevně zasunut a zda nedošlo ke zkratu nebo rozpojení;</li> <li>Pokud se závada neodstraní, opravte nebo vyměňte sestavu magnetického editoru.</li> </ol>
40	Porucha modulu magnetického snímače - selhání kalibrace nuly	<ol> <li>I. Znovu vynulujte sestavu magnetického opletení;</li> <li>Pokud se závada neodstraní, opravte nebo vyměňte sestavu magnetického editoru.</li> </ol>
41	Porucha modulu magnetického snímače - porucha s více otáčkami	<ol> <li>Zkontrolujte, zda není magnetické opletení a upevňovací šroub hřídele motoru uvolněný;</li> <li>Pokud se závada neodstraní, opravte nebo vyměňte sestavu magnetického editoru.</li> </ol>
42	Porucha modulu magnetického sníma víceotáčkový malý magnetický sn nedošlo k poruše víceotáčkového sn magnetického opletení čipu je	nče - ímač1 . Zkontrolujte, zda ímače malého abnormální; 2. Pokud se závada neodstraní, opravte nebo vyměňte sestavu
43	Porucha modulu magnetického sníma víceotáčkový velký magnetický sn k poruše víceotáčkového snímače abnormální;	magnetického editoru. iče - úmač 1. Zkontrolujte, zda nedošlo magnetického opletení čipu je 2. Pokud se závada neodstraní, opravte nebo vyměňte sestavu magnetického editoru.

Tabulka	2 -	pokračováníMáma	předchozí	strana
rabuilla	2	politideovarinitiarita	predenozi	Suana

Kód závady	Název poruchy	Způsob zpracování
44	Porucha modulu magnetického opletení - porucha jednootáčkového magnetického opletení	<ol> <li>Zkontrolujte, zda není abnormální jednootáčkový čip magnetického opletení;</li> <li>Pokud se závada neodstraní, opravte nebo vyměňte sestavu magnetického editoru.</li> </ol>
45	Porucha modulu magnetického snímače - porucha optického snímače	<ol> <li>I. Zkontrolujte, zda je optický kódovací disk znečištěný nebo zda není zaseknutý;</li> <li>Pokud se závada neodstraní, opravte nebo vyměňte sestavu magnetického editoru.</li> </ol>

#### Tabulka 2 - pokračování Máma předchozí strana

### 1.3.7.3 Dodatek 3: Koncová deska 485 upgrade

Během používání v terénu je možné firmware aktualizovat tak, aby splňoval nové požadavky. K aktualizaci svorkovnice přes rozhraní 485 (je vyžadován mod- ule USB na 485) bude poskytnut nový aktualizační soubor (XX\_XX\_MAIN. bin). Kroky aktualizace jsou následující:

**Krok 1: Zapojení 485.** Na konci robota je 5pinový komunikační letecký konektor. Rozložení pinů a popis pinů leteckého konektoru jsou uvedeny na obr. 1. Propojte 485 - A a 485 - B na konci robota s A a B nástroje USB na 485 pomocí kroucených dvojlinek.



Obrázek 6.3-1 Rozložení pinů leteckého konektoru

Krok 2: hardwarové připojení. připojte konec USB nástroje USB-485 k počítači a ve správci zařízení počítače, pokud je nástroj USB-485 identifikován, se zobrazí následující rozhraní.

Obrázek 6.3-2 Identifikace portů USB&485 Popis

**Krok3Aktualizujte nástroj.** Po dokončení zapojení otevřete "MSerial Port Debugging Assistant", klikněte na tlačítko "Terminal Board" a ve funkci "Serial Port Parameter Setting" vyberte výše identifikovaný sériový port. Přenosová rychlost je 115200, datový bit je 8, kontrolní bit je žádný a stop bit je 1. Poté otevřete sériový port. Po úspěchu se zobrazí výzva "Serial port opened successfully" (Sériový port byl úspěšně otevřen). Intel(R) Active Management Technology - SOL (COM3)
 JLink CDC UART Port (COM4)
 USB-SERIAL CH340 (COM2)

xe:	rialportse	
port:	COM2	
baudrate	115200	*
databit:	B	
ere:	NON	¥
endbit:	1)	
e	Lose port	

Obrázek 6.3-3 Nastavení parametrů sériového portu

Krok4Aktualizace firmwaru. Vyberte "End plate" a klikněte na "Firmware upgrade", jak je znázorněno na obrázku.

trlbord	servobord	slezbord	waxber d				
se	rialportset		re	ovdataarea		naintaininfo producttest firmupgrade	
port:	CON3 +						
beudrate	115290 *						
databit:	B					Interest	
ere:	вон +						
endbit:	‡ =						
e	lase part						
	dataret	-				alexfireversion.	
V Next	Sand						
- Head	Recv					0	
🗐 auto	reand					readsyslog	
104	1	6		senddata			
in auto							
10s	*				Lenders	Ecutinfo	
8	5.55 5.55	- 1995 - 1985	operation —	14.0			
1000000	lear recv	clear stops	thos clear	zavedata	about		

Obrázek 6.3-4 Aktualizace firmwaru koncové desky

- Nejprve klikněte na možnost "Flash Erase". Po úspěšném vymazání se v oblasti pro příjem dat zobrazí výzva, že vymazání proběhlo úspěšně.
- Otevřete soubor (soubor, který má být aktualizován) a vyberte cestu k jeho uložení, jak je znázorněno níže. Po výběru se v poli pro zobrazení názvu souboru objeví název souboru, který má být upgradován.



Obrázek 6.3-5 Výběr aktualizačního souboru

 Klikněte na tlačítko "Odeslat soubor", a když se na ukazateli průběhu zobrazí 100 %, znamená to, že soubor aktualizace byl odeslán.

**Ověření aktualizace Step5Upgrade.** Systém je restartován a zapnut. Ve sloupci "Maintenance Information" (Informace o údržbě) vyberte možnost "Query terminal board firmware version information" (Dotaz na informace o verzi firmwaru svorkovnice) a informace o verzi firmwaru se zobrazí v "Receive Data Area" (Oblast příjmu dat). Pokud se shoduje s informacemi o verzi aktualizovaného souboru, je aktualizace úspěšná, v opačném případě se aktualizace nezdaří.

14			Francesc Interprese	
			unicast	
			alexfiraversion	
			readsyslog	
	senddata	xenddata		

Obrázek 6.3-6 Dotazování na informace o verzi firmwaru

### 1.3.7.4 Dodatek 4: Upgrade řídicí jednotky 485

Na desce řídicí jednotky robota je rozhraní "power communication" a nástroje USB&485 A a B jsou připojeny k rozhraní "485-A" a "485-B".

Postup aktualizace je stejný jako u svorkovnice a podle toho lze zvolit software, který zde nebudeme opakovat.

Obrázek 6.4-1 Komunikační rozhraní pro napájení

### 1.3.7.5 Dodatek 5: Seznam náhradních dílů a zranitelných dílů

Název náhradního dílu	Díl č.	Množství/kus(y)
Šroub M8 * 30	4.0.08.2006185	4
Přímý kolík typu A 8 * 20	4.5.00.2013076	2
Pojistka 5x20 6A	/	1



### 1.3.8 Termín

### Kategorie vypnutí

- Vypnutí třídy 0Při odpojení napájení robota přestane robot okamžitě pracovat. Jedná se o nekontrolovatelné zastavení. Protože každý kloub bude brzdit nejrychlejší rychlostí, může se robot odchýlit M om dráhu nastavenou programem. toto ochranné zastavení lze použít při překročení limitu bezpečnostního vyhodnocení nebo při chybě v části bezpečnostního vyhodnocení řídicího systému. Další informace naleznete v normě ENISO13850:2008 nebo IEC60204-1:2006.
- Vypnutí třídy 1Pokud je robot napájen na zastavení, robot se zastaví. Když se robot zastaví, dojde k
  odpojení napájení. Jedná se o řízené zastavení a robot bude následovat naprogramovanou dráhu. odpojte
  napájení po jedné sekundě nebo jakmile se robot stabilně zastaví. Další informace naleznete v normě
  ENISO13850:2008 nebo IEC60204-1:2006.
- **Vypnutí třídy** 2Kontrolovatelné zastavení při zapnutí robota. Robot zastaví všechny pohyby během jedné sekundy.Provoz řídicího systému bezpečnostního vyhodnocení může způsobit, že robot zůstane v poloze zastavení. Další informace naleznete v normě IEC60204-1:2006.

**Míra pokrytí diagnostikouDCPoužívá se k** měření účinnosti diagnostiky zavedené za účelem dosažení hodnocené úrovně výkonnosti. Další informace naleznete v normě ENISO13849-1:2008.

Integrátor Integrátor je mechanismus, který navrhuje konečnou instalaci robota. Integrátor je odpovědný za konečné posouzení rizik a musí zajistit, aby konečná instalace byla v souladu s místními zákony a předpisy.

**Střední doba do nebezpečné** poruchyMTTFdStřední doba do nebezpečné poruchy (MTTFd) označuje hodnotu vypočtenou a zjištěnou pro dosažení posuzované úrovně výkonu. Další informace naleznete v normě ENISO13849-1:2008.

Posouzení rizikPosouzení rizik je celý proces identifikace všech rizik a jejich snížení na vhodnou úroveň.Posouzení rizik se zaznamenává a archivuje.Podrobnosti naleznete v normě ISO12100.

**Úroveň** výkonnostiÚroveň výkonnosti (PL) je samostatná úroveň, která se používá k popisu schopnosti každé části řídicího systému související s bezpečností vykonávat bezpečnostní funkce za předvídatelných podmínek. PL je druhou nejvyšší kategorií spolehlivosti, což znamená, že bezpečnostní funkce je poměrně důvěryhodná.více informací naleznete v normě EN ISO13849-1:2008.

Spojovací přírubaKonstrukce sloužící ke spojení vnějších nástrojů, obecně nazývaná příruba.

Konec robota Středový bod poslední osy nebo spojovací příruby robota.

**Středový bod** nástrojeTCTCStředový bod nástroje je charakteristický bod robotického nástroje, který je řídicím bodem robotického systému. Při opuštění výrobního závodu je výchozí hodnotou střed poslední pohyblivé osy nebo spojovací příruby. Středový bod každého nástroje obsahuje transformace a rotace nastavené vzhledem ke středu výstupní příruby nástroje. Souřadnice polohy X, Y, Z určují polohu středového bodu nástroje a RX, RY, RZ určují směr středu nástroje. když jsou všechny hodnoty nulové, střed nástroje se shoduje se středem spojovací příruby.

**Poloha nástroje a polohový bodTCPodává** polohu souřadnicového systému nástroje vůči souřadnicovému systému koncového článku na základě středového bodu nástroje TCP.

**Základní souřadnicový** systémPočátek základního souřadnicového systému je obecně definován ve středovém bodě mezi první osou robota a montážním povrchem. Osa x směřuje dopředu a osa y je určena podle pravidla pravé ruky v axiálním směru.

**Světový souřadnicový** systémPevný souřadnicový systém vytvořený v pracovní jednotce nebo na pracovišti.Pokud existuje pouze jeden robot, lze tento souřadnicový systém považovat za shodný se základním souřadnicovým systémem;Pokud existuje více robotů nebo externích zařízení, může světový souřadnicový systém poskytovat jedinečný referenční systém pro tato zařízení. Za předpokladu, že souřadnicový systém vyhovuje pro pohodlnou kalibraci ostatních zařízení, lze jeho konkrétní polohu libovolně specifikovat.

**Kloubní souřadnicový systém** Kloubní souřadnicový systém je souřadnicový systém v kloubu robota. V kloubovém souřadnicovém systému může každá osa robota dosáhnout nezávislého pohybu vpřed nebo vzad v mezním rozsahu. Je použitelný pro robota, který se potřebuje pohybovat ve velkém rozsahu a nevyžaduje polohu robota TCP. Pohyb robota v jedné ose v ručním režimu se provádí v kloubovém souřadnicovém systému.

**Souřadnicový systém** nástrojeSouřadnicový systém používaný k určení polohy středového bodu nástroje a polohy nástroje. Pokud není definován, je souřadný systém nástroje ve výchozím nastavení středem připojovací příruby. po instalaci nástroje se TCP změní a stane se středem konce nástroje.

**Externí souřadnicový systém nástroje**: Souřadnicový systém používaný k určení polohy a pozice nástrojů upevněných mimo robota.

**Prodloužení** osyOdstranění osy na těle robota a přidání další osy pro práci. Rozšiřující osa zahrnuje především posuvnou lištu, otočný stůl a externí servořízení.

Manuální režim V tomto režimu jsou všechny pohyby robota řízeny ručně uživatelem a nefungují vnější bezpečnostní zařízení, jako jsou bezpečnostní mříže a bezpečnostní dveře, aby se usnadnilo blízké ladění.

Automatický režimTento režim se obvykle používá pro spuštění výukového programu robota. V tomto okamžiku jsou povolena externí bezpečnostní zařízení.

**Přesnost opakovaného** polohováníSoulad polohy a polohy naměřené robotem ve stejných podmínkách a stejnou metodou n-krát.

Výukový přívěsekRuční jednotka, která programuje nebo pohybuje robotem a je připojena k řídicímu systému.

### KAPITOLA

# DVA

# SDK MANUÁL

# 2.1 C++

Tato příručka je sekundárním dokumentem vývojového rozhraní jazyka C++.

**Důležité:** Popis jednotky parametrů robota: Jednotkou polohy robota je milimetr (mm) a jednotkou polohy je stupeň (°).

### Důležité:

- 1) V příkladech kódu, které nejsou výslovně uvedeny, byl robot ve výchozím nastavení zapnut a povolen;
- 2) Všechny příklady kódu v dokumentaci jsou ve výchozím nastavení bez zásahu do pracovního prostoru robota;
- 3) Při zkoušce skutečného použití použijte údaje robota na místě.

### 2.1.1 Specifikace datové struktury

### 2.1.1.1 Typ návratové hodnoty volání rozhraní

typedef int errno\_t;

1

### 2.1.1.2 Typ dat o společné poloze

```
1 {/**
2 @brief Datový typ společné polohy
3 */
4 typedef struct
5 {
6 double jPos[6]; /* Šest pozic kloubů, jednotka: deg */
7 {JointPos;
```

### 2.1.1.3 Kartézský typ prostorových dat o poloze

1	/**			
2	* @krátký kartéz	zský prostorový datový typ		
3	polohy			
4	*/			
5	typedef struct			
6	<pre>{ double x;</pre>	/* Souřadnice osy X, jednotka:	*/	
7	double y;	mm	*/	
8	double z;	/* Souřadnice osv Y. jednotka:	*/	
9	} DescTran;	mm		
		/* souřadnice osy Z, jednotka:		

### 2.1.1.4 Datový typ Eulerova úhlu

mm

```
/**
1
   * @krátký datový typ Eulerova úhlu
2
  polohy
3
   */
4
  typedef struct
5
       double rx;
                    /* Úhel natočení kolem pevné osy X, jednotka:
                                                                        */
6
   {
       double ry;
                                                                        */
                    deg
7
       double rz;
                                                                        */
                    /* Úhel natočení kolem pevné osy y, jednotka:
8
  } Rpy;
                    deg
9
                    /* Úhel natočení kolem pevné osy Z, jednotka:
                    deg
```

### 2.1.1.5 Datový typ polohy v kartézském prostoru

```
/**
1
   *@krátký typ pozice v kartézském
2
  prostoru
3
   */
4
  typedef struct
5
                            /* kartézská poloha
                                                     */
   {
6
                            /* Postoj v kartézském prostoru
7
       DescTran tran;
       Rpy rpy;
                                                            */
8
  } DescPose;
```

### 2.1.1.6 Datový typ polohy prodlužovací osy

### 2.1.1.7 Typ dat snímače točivého momentu

1	/**	
2	* @brief Složka .	síly a složka točivého momentu snímače síly
3	*/	
4	typedef struct	
5	{	
6	double	/* Složka síly podél osy x, jednotka: N */
7	IX; double	/* Složka síly podél osy y, jednotka: N */
8	fr: double	/* Složka sílv podél osv z. jednotka: N */
9	ty: double	/* Složka točivého momentu kolem osy X, jednotka: Nm */
10	ty: double	/* Složka točivého momentu kolem osy Y, jednotka: Nm
11	tz:	*/
12	} ForceTorque;	/* Složka točivého momentu kolem osy Z, jednotka: Nm
		*/

### 2.1.1.8 Datový typ parametru spirály



### 2.1.2 Základy

### 2.1.2.1 Instanciovat robota

```
1 /**
2 * Konstruktor třídy rozhraní @brieMobot
3 */
4 MRobot();
```

6

### 2.1.2.2 naváže komunikaci s řídicí jednotkou

/\*\* 1 \* @brief Navázání komunikace s řídicí jednotkou robota 2 \* @param [in] ip IP adresa řadiče. Výchozí hodnota je 192.168.58.2 3 4 \* @*return Kód chyby* 5 \*/ errno\_t RPC(const char \*ip);

### 2.1.2.3 Dotaz na číslo verze SDK

```
/**
1
  * @brief Dotaz na číslo verze SDK
2
  * @param [out] verze
3
                         Verze SDK
4
  * @return Kód chyby
  */
5
6
  errno_t__GetSDKVersion(char *version);
```

### 2.1.2.4 Získání IP adresy řadiče

/\*\* 1 \* @brief Získání IP adresv řadiče 2 \* @param [out] ip IP adresa řadiče 3 4 \* @*return* Kód chyby 5 \*/ 6 errno\_t\_GetControllerIP(char \*ip);

### 2.1.2.5 Ovládání robota pro vstup do režimu výuky tažením nebo výstup z něj.

```
/**
1
   * @brief Ovládání robota pro vstup do režimu výuky tažením nebo výstup z něj
2
3
  * @param [in] stav 0-výstup z režimu přetahování1-vstup do režimu přetahování
4
  * @return Kód chyby
5
  */
6
  errno_t DragTeachSwitch(uint8 t state);
```

### 2.1.2.6 Dotazuje se, zda je robot v režimu přetahování

```
/**
1
   * @brief Kontrola, zda je robot v režimu přetahování
2
3
  * @param [out] stav 0-nevyužití výukového režimu1-využití výukového režimu
4
   * @return Kód chyby
5
   */
6
  errno_t IsInDragTeach(uint8_t *state);
```

### 2.1.2.7 Povolení ovládání nahoru a dolů

6

1

2

4

5

6

/\*\*

funkce\_\_ ↔ je aktivován automaticky po zapnutí robota.

\* @brief Povolení nebo zakázání funkce na robotu nebo mimo něj. Ve výchozím nastavení je

\* @param [in] stav 0-down-enable1-upper enable

\* @return Kód chyby

errno\_t RobotEnable(uint8\_t state);

### 2.1.2.8 Ruční/automatický režim ovládání robota

```
/**

* @brief Ovládání robota v ručním/automatickém režimu

* @param [in] režim 0-automatický režim1-ruční režim

* @return Kód chyby

*/
errno_t Mode(int mode);
```

### 2.1.2.9 Příklad kódu

```
#include <cstdlib>
1
   #include <iostream>
2
   #include <stdio.h>
3
   #include <cstring>
4
   #include <unistd.h>
5
   #include "MRobot.h"
6
   #include "RobotTypes.h"
7
8
   using namespace std;
9
10
   int main(void)
11
   {
12
        MRobot robot;
                                          //Instituce objektu robota
13
        robot.RPC("192.168.58.2");
                                          /Navázat komunikační spojení s robotem
14
    ⊶kontrolér
15
        char ip[64]="";
16
        char version[64] = "";
17
        uint8 t state;
18
19
        robot.GetSDKVersion(version);
20
        printf("SDK version:%s\n", version);
21
        robot.GetControllerIP(ip);
22
        printf("controller ip:%s\n", ip);
23
24
        robot.Mode(1);
25
        sleep(1);
26
        robot.DragTeachSwitch(1);
27
        robot.IsInDragTeach(&state);
28
```

```
printf("drag state :%u\n", state);
29
        sleep(3); robot.DragTeachSwitch(0);
30
        sleep(1);
31
        robot.IsInDragTeach(&state);
32
        printf("drag state :%u\n", state);
33
        sleep(3);
34
35
        robot.RobotEnable(0);
36
        sleep(3);
37
        robot.RobotEnable(1);
38
39
        robot.Mode(0);
40
        sleep(1);
41
        robot.Mode(1);
42
        vrátit 0;
43
44
45
   }
46
```

## 2.1.3 Pohyb

### 2.1.3.1 Pohyb bodu Jog

```
/**
1
   * abrief Pohyb bodu Jog
2
   * @param [in] ref 0- pohyb uzlu, 2- základní souřadnicový systém, 4- souřadnice
3
     nástroje
4
     systém, 8- souřadnicový systém obrobku
   * @param [in] nb 1-kloub 1(nebo osa x), 2-kloub 2(nebo osa y), 3-kloub 3(nebo osa z),
   *-@hallan(neblo rolince-hogentivniyko);ekdoab15p(azibioniskarelaokewelosy y), kloub 6(nebo_).
5
   *- @potate [kdlemPosse a) to rychlosti, [0~100]
6
   * @param [v]
7
                    acc Procento zrychlení, [0~100]
8
   * @param [v]
                    max_dis Maximální úhel jednoho kliknutí, jednotka: [°] nebo vzdálenost,
   * @return Kód chybyotka: [mm]
9
   */
10
   errno_t StartJOG(uint8_t ref, uint8_t nb, uint8_t dir, float vel, float acc, float max_
11
   , →dis);
```

### 2.1.3.2 Zastavení dynamického zpomalení v bodě Jog

```
Kapitola 2. Příručka
SDK
```

### 2.1.3.3 Běh se okamžitě zastaví

/\*\* \* @brief Běh se okamžitě zastaví \* @return Kód chyby \*/ errno t ImmStopJOG();

1

2

4

### 2.1.3.4 Příklad kódu

```
#include <cstdlib>
1
   #include <iostream>
2
   #include <stdio.h>
3
   #include <cstring>
4
   #include <unistd.h>
5
   #include " MRobot.h"
6
   #include "RobotTypes.h"
7
8
   using namespace std;
9
10
   int main(void)
11
   {
12
       Robot MRobot;
                                          //Instituce objektu robota
13
       robot.RPC("192.168.58.2");
                                          /Navázání komunikačního spojení s robotem
14
   ⊶kontrolér
15
       robot.StartJOG(0,1,0,20.0,20.0,30.0); //Pro pohyb v jednom kloubu je StartJOG ne-
16
   ↔ příkaz blokování. Přijímání jiných pohybových příkazů (včetně StartJOG) během pohybu
    \leftrightarrow_{je} vv \check{r}azeno
        spánek(1);
17
       //robot.StopJOG(1) //Zastavení zpomalení jedné osy robota
18
       robot.ImmStopJOG(); //Jediná osa robota se okamžitě zastaví
19
       robot.StartJOG(0,2,1,20.0,20.0,30.0);
20
       sleep(1);
21
       robot.ImmStopJOG();
22
       robot.StartJOG(0,3,1,20.0,20.0,30.0);
23
       sleep(1);
24
       robot.ImmStopJOG();
25
       robot.StartJOG(0,4,1,20.0,20.0,30.0);
26
       spánek(1);
27
       robot.ImmStopJOG();
28
       robot.StartJOG(0,5,1,20.0,20.0,30.0);
29
       sleep(1);
30
       robot.ImmStopJOG();
31
       robot.StartJOG(0,6,1,20.0,20.0,30.0);
32
       spánek(1);
33
       robot.ImmStopJOG();
34
35
       robot.StartJOG(2,1,0,20.0,20.0,30.0); //Bod v základním souřadném systému
36
       spánek(1);
37
       //robot.StopJOG(3) //Zastavení zpomalení jedné osy robota
38
```

```
(pokračování na předchozí straně)
```

```
robot.ImmStopJOG(); //Jediná osa robota se okamžitě zastaví
39
        robot.StartJOG(2,2,1,20.0,20.0,30.0);
40
       spánek(1);
41
        robot.ImmStopJOG();
42
        robot.StartJOG(2,3,1,20.0,20.0,30.0);
43
       spánek(1);
44
       robot.ImmStopJOG();
45
       robot.StartJOG(2,4,1,20.0,20.0,30.0);
46
        spánek(1);
47
       robot.ImmStopJOG();
48
       robot.StartJOG(2,5,1,20.0,20.0,30.0);
49
        sleep(1):
50
        robot.ImmStopJOG();
51
       robot.StartJOG(2,6,1,20.0,20.0,30.0);
52
       spánek(1);
53
       robot.ImmStopJOG();
54
55
       robot.StartJOG(4,1,0,20.0,20.0,30.0); //Bod v souřadném systému nástroje
56
        spánek(1);
57
        //robot.StopJOG(5) //Zastavení zpomalení jedné osy robota
58
       robot.ImmStopJOG(); //Jediná osa robota se okamžitě zastaví
59
       robot.StartJOG(4,2,1,20.0,20.0,30.0);
60
       sleep(1):
61
       robot.ImmStopJOG();
62
       robot.StartJOG(4,3,1,20.0,20.0,30.0);
63
       sleep(1);
64
        robot.ImmStopJOG();
65
       robot.StartJOG(4,4,1,20.0,20.0,30.0);
66
       spánek(1);
67
        robot.ImmStopJOG();
68
       robot.StartJOG(4,5,1,20.0,20.0,30.0);
69
        sleep(1);
70
       robot.ImmStopJOG();
71
       robot.StartJOG(4,6,1,20.0,20.0,30.0);
72
       sleep(1);
73
       robot.ImmStopJOG();
74
75
       robot.StartJOG(8,1,0,20.0,20.0,30.0); //Bod v souřadném systému obrobku
76
       sleep(1);
//robot.StopJOG(9) //Zastavení zpomalení jedné osy robota
77
78
       robot.ImmStopJOG(); //Jediná osa robota se okamžitě zastaví
79
       robot.StartJOG(8,2,1,20.0,20.0,30.0);
80
       sleep(1);
81
       robot.ImmStopJOG();
82
       robot.StartJOG(8,3,1,20.0,20.0,30.0);
83
       spánek(1);
84
       robot.ImmStopJOG();
85
       robot.StartJOG(8,4,1,20.0,20.0,30.0);
86
        spánek(1);
87
        robot.ImmStopJOG();
88
       robot.StartJOG(8,5,1,20.0,20.0,30.0);
89
        sleep(1);
90
```

```
91 robot.ImmStopJOG();
92 robot.StartJOG(8,6,1,20.0,20.0,30.0);
93 sleep(1);
94 robot.ImmStopJOG();
95 vrátit 0;
97 }
```

### 2.1.3.5 Pohyb v kloubním prostoru

```
/**
1
   * @brief Pohyb v kloubním prostoru
2
   * @param [in] joint pos Poloha cílového kloubu,
3
   * aparam [in]ndaka: pose Cílová kartézská poloha
4
   * @param [in] tool Číslo souřadnice nástroje, rozsah [1~15] Číslo
5
   * @param [in] user souřadnice obrobku, rozsah [1~15] Procento
6
   * @param [in] vel
7
                        otáček, rozsah [0~100]
   * @param [in] acc
8
                        Procento zrychlení, rozsah [0~100], zatím neotevřeno
   * @param [in] ovl
9
   * @param [in] epos Faktor škálování rychlosti, rozsah[0~100]
10
   * @param [in] blendT P-d!Ohapotzptěrnérhůšíd eleloje skrátkja f 044500.0]- doba vyhlazování,
11
   →(neblokující), v ms
   * @param [in] offset flag
                                 0- žádný offset, 1- offset v základním/pracovním souřadném
12
    systému, 2-
13
   ·→ posun v souřadném systému nástroje
14
   * (aparam [in] offset pos
                                Posunutí pózy
15
   * @return Kód chyby
16
   */
   errno_t MoveJ(JointPos *joint_pos, DescPose *desc_pos, int tool, int user, float vel,___
   -float acc, float ovl, ExaxisPos *epos, float blendT, uint8 t offset_flag, DescPose,
   \leftrightarrow*offset_pos);
```

2.1.3.6 Přímočarý pohyb v kartézském prostoru

1	/**
2	* @brieMectilineární pohyb v kartézském
3	program [in] joint_pos Poloha cílového kloubu,
4	* @param Jealpoteka:polsg Cílová kartézská poloha
5	* @param [in] tool Číslo souřadnice nástroje, rozsah [1~15] Číslo
6	* @param [in] user souřadnice obrobku, rozsah [1~15] Procento
7	* @param [in] vel otáček rozsah [0-100]
8	* @param [in] acc
9	* @param [in] ovl Procento zrychleni, rozsah [0~100], zatim neotevreno
10	* @param [in] blend R [141:0] Škólaytán hrvák losti diesech [9, [00] b00.0] - poloměr vyhlazování
	→(neblokující), jednotka: mm
11	* @param [in] epos Poloha rozpěrné hřídele, jednotka: mm
12	* @param [in] search 0- bez vyhledávání drátů, 1- vyhledávání drátů
13	* @param [in] offset_flag 0- žádný offset, 1- offset v základním/pracovním souřadném
14	systému, 2
14	↔ posun v souřadném systému nástroje
	* @param [in] offset pos Posunutí pózy

\* @return Kód chyby

(pokračování na předchozí straně)

15 16 17

\*/ errno\_t MoveL(JointPos \*joint\_pos, DescPose \*desc\_pos, int tool, int user, float vel, \_\_\_\_\_ ·→float acc, float ovl, float blendR, ExaxisPos \*epos, uint8\_t search, uint8\_t offset\_ \_\_\_\_\_\_, DescPose \*offset\_pos);

### 2.1.3.7 Pohyb po kruhovém oblouku v kartézském prostoru

1	/**	
2	* @brief Pohyb po kruhovém oblouku v kartézském prostoru	
3	* @param [in] joint_pos_p Poloha kloubu waypointu, jednotka: deg	
4	* @param [in] desc_pos_p Kartézská poloha bodu cesty	
5	* @param [in] ptool Souřadnicové číslo nástroje, rozsah [1~15]	
6	* @param [in] puser Souřadnicové číslo obrobku, rozsah [1~15]	
7	* @param [in] pvel Procento rychlosti, rozsah [0~100]	
8 9	* @param [in] pacc Procento zrychlení, rozsah [0~100], zatím neotevřeno * @param [in] epos_p Poloha rozpěrné hřídele, jednotka: mm	
10	* @param [in] poffset_flag 0- žádný offset, 1- offset v základním/pracovním souřadném sys ↔ posun v souřadném systému nástroje	stému, 2
11	* @param [in] offset_pos_p Posunutí pozice	
12	* @param [in] joint_pos_t Cílová poloha kloubu, jednotka: deg	
13	* @param [in] desc_pos_t Kartézská poloha cílového bodu	
14	* @param [in] ttool Číslo souřadnice nástroje, rozsah [1~15]	
15	* @param [in] tuser Souřadnicové číslo obrobku, rozsah [1~15]	
16	* @param [in] tvel Procento rychlosti, rozsah [0~100]	
17 18	* @param [in] tacc Procento zrychlení, rozsah [0~100], zatím neotevřeno * @param [in] epos_t Poloha rozpěrné hřídele, jednotka: mm	
19	* @param [in] toffset_flag 0- žádný offset, 1- offset v základním/pracovním souřadném sys ↔ posun v souřadném systému nástroje	stému, 2
20 21	* @param [in] offset_pos_t Posunutí pozice * @param [in] ovl Faktor škálování rychlosti, rozsah[0~100]	
22	* @param [in] blendR [-1.0]- pohyb na místě (blokování), [0~1000.0]- poloměr vyhlazování, -(neblokující), jednotka: mm	
23 24	* @return Kód chyby */	
25	errno_t MoveC(JointPos *joint_pos_p, DescPose *desc_pos_p, int ptool, int puser, float_ →pve/, float pacc, ExaxisPos *epos_p, uint8_t poffset_flag, DescPose *offset_pos_p, →JointPos *joint_pos_t, DescPose *desc_pos_t, int ttool, int tuser, float tvel, float_ →tecc, ExaxisPos *epos_t, uint8_t toffset_flag, DescPose *offset_pos_t, float ovl, float_	
	→blendR);	

### 2.1.3.8 Kruhový pohyb v kartézském prostoru

```
/**
1
   * @brief Kruhový pohyb v kartézském prostoru
2
   * @param [in] joint pos p Poloha kloubu bodu cesty 1, jednotka: deg
3
   * @param [in] desc pos p
                               Kartézská poloha bodu 1
4
   * @param [in] ptool
                       Souřadnicové číslo nástroje, rozsah [1~15]
5
   * @param [in] puser
                         Souřadnicové číslo obrobku, rozsah [1~15]
6
                        Procento rychlosti, rozsah [0~100]
   * @param [in] pvel
7
                        Procento zrychlení, rozsah [0~100], zatím neotevřeno
   * @param [in] pacc
8
   * @param [in] epos p Poloha rozpěrné hřídele, jednotka: mm
9
   * @param [in] joint pos t
                               Kloubová poloha v bodě 2, jednotka: deg
10
   * @param [in] desc pos t
                               Kartézská poloha bodu trasy 2
11
   * @param [in] ttool
                         Číslo souřadnice nástroje, rozsah [1~15]
12
   * @param [in] tuser
                         Souřadnicové číslo obrobku, rozsah [1~15]
13
                        Procento rychlosti, rozsah [0~100]
   * @param [in] tvel
14
                        Procento zrychlení, rozsah [0~100], zatím neotevřeno
   * @param [in] tacc
15
   * @param [in] epos t Poloha rozpěrné hřídele, jednotka: mm
16
   * @param [in] ovl
                       Faktor škálování rychlosti, rozsah[0~100]
17
                               0- žádný offset, 1- offset v základním/pracovním souřadném systému, 2-
   * @param [in] offset flag
18
   ↔ posun v souřadném systému nástroje
   * @param [in] offset pos
                              Posunutí pozice
19
   * @return Kód chvbv
20
   */
21
   errno_t Circle(JointPos *joint_pos_p, DescPose *desc_pos_p, int ptool, int puser, float_
22
   →pvel, float pacc, ExaxisPos *epos_p, JointPos *joint_pos_t, DescPose *desc_pos_t, int_
```

### *→flag*, DescPose \*offset\_pos);

### 2.1.3.9 Příklad kódu

```
#include <cstdlib>
1
   #include <iostream>
2
   #include <stdio.h>
3
   #include <cstring>
4
   #include <unistd.h>
5
   #include "MRobot.h"
6
   #include "RobotTypes.h"
7
8
   using namespace std;
9
10
   int main(void)
11
   ł
12
                                           //Instituce objektu robota
        MRobot robot:
13
        robot.RPC("192.168.58.2");
                                           /Navázat komunikační spojení s robotem
14
      →kontrolér
15
        JointPos j1, j2, j3, j4;
16
        DescPose desc_pos1,desc_pos2,desc_pos3,desc_pos4,offset_pos;
17
        ExaxisPos epos;
18
19
                                                                                      (pokračování na další straně)
```

20 21 22 23 24 25 26 27 28	<pre>memset(&amp;j1, 0, sizeof(JointPos)); memset(&amp;j2, 0, sizeof(JointPos)); memset(&amp;j3, 0, sizeof(JointPos)); memset(&amp;j4, 0, sizeof(JointPos)); memset(&amp;desc_pos1, 0, sizeof(DescPose)); memset(&amp;desc_pos2, 0, sizeof(DescPose)); memset(&amp;desc_pos3, 0, sizeof(DescPose)); memset(&amp;desc_pos4, 0, sizeof(DescPose)); memset(&amp;coffset_pos, 0, sizeof(DescPose));</pre>
29 30	memset(&epos, 0, sizeof(ExaxisPos));
<ul> <li>31</li> <li>32</li> <li>33</li> <li>34</li> <li>35</li> <li>36</li> <li>37</li> <li>28</li> </ul>	<pre>j1 = {114.578,-117.798,-97.745,-54.436,90.053,-45.216}; desc_posl.tran.x = -140.418; desc_posl.tran.y = 619.351; desc_posl.tran.z = 198.369; desc_posl.rpy.rx = -179.948; desc_posl.rpy.ry = 0.023; desc_posl.rpy.rz = 69.793;</pre>
38 39	j2 = {121.381,-97.108,-123.768,-45.824,89.877,-47.296};
40 41 42 43	desc_pos2.tran.x = -127.772; desc_pos2.tran.y = 459.534; desc_pos2.tran.z = 221.274; desc_pos2.rpy.rx = -177.850;
44	desc_pos2.rpy.ry = -2,507;
45 46	desc_pos2.rpy.rz = 78.627;
47 48 49 50 51 52	<pre>j3 = {138.884,-114.522,-103.933,-49.694,90.688,-47.291}; desc_pos3.tran.x = -360.468; desc_pos3.tran.y = 485.600; desc_pos3.tran.z = 196.363; desc_pos3.rpy.rx = -178.239; desc_pos3.rpy.ry = -0,893;</pre>
53	desc_pos3.rpy.rz = 96.172;
55 56 57 58 59 60 61 62	j4 = {159.164,-96.105,-128.653,-41.170,90.704,-47.290}; desc_pos4.tran.x = -360.303; desc_pos4.tran.y = 274.911; desc_pos4.tran.z = 203.968; desc_pos4.rpy.rx = -176.720; desc_pos4.rpy.ry = -2.514; desc_pos4.rpy.rz = 116.407;
62 63 64 65 66 67 68 69 70 71	<pre>int tool = 0; int user = 0; float vel = 100.0; float acc = 100.0; float ovl = 100.0; float blendT = 0,0; float blendR = 0.0; uint8_t flag = 0; uint8_t search = 0;</pre>

```
72
       robot.SetSpeed(20);
73
74
       int err1 = robot.MoveJ(&j1, &desc_pos1, tool, user, vel, acc, ovl, &epos, blendT,
75
   , &offset_pos);
       printf("movej errcode:%d\n", err1);
76
77
       int err2 = robot.MoveL(&j2, &desc_pos2, tool, user, vel, acc, ovl, blendR, &epos,
78
   errcode:%d\n", err2);
79
80
       int err3 = robot.MoveC(&j3,&desc_pos3,tool,user,vel,acc,&epos,flag,&offset_pos,&j4,&
81
   →desc_pos4, tool, user, vel, acc, & epos, flag, & offset_pos, ovl, blendR);
       printf("movec errcode:%d\n", err3);
82
83
       int err4 = robot.MoveJ(&j2, &desc_pos2, tool, user, vel, acc, ovl, &epos, blendT,
84
   →flag, &offset_pos);
       printf("movej errcode:%d\n", err4);
85
86
       int err5 = robot.Circle(&j3,&desc_pos3,tool,user,vel,acc,&epos,&j4,&desc_pos4,tool,
87
    --user, vel, acc, & epos, ovl, flag, & offset_pos);
        printf("errcode kruhu:%d\n", err5);
88
89
       vrátit 0;
90
   }
91
```

### 2.1.3.10 Spirálový pohyb v kartézském prostoru

1	/**
2	* @brief Spirálový pohyb v kartézském prostoru
3	* @param [in] joint pos Poloha cílového kloubu,
4	* @param fiedholaka:pasg Cílová kartézská poloha
5	* @param [in] tool Číslo souřadnice nástroje, rozsah [1~15] Číslo
6	* @param [in] user souřadnice obrobku, rozsah [1~15] Procento
7	* @param [in] vel otáček rozsah [0~100]
8	* aparam [in] acc Procento zwichlani rozsah [0, 100] zatim neotewizano Poloha
9	* $(aparam [in] epos$ 170cento 21ychient, 702sun $[0~700]$ , 2000 neolevieno 1010nu
10	* aparam [in] ovl rozperneho hridele, jednotka: mm
11	* @param [in] offset <b>[flalg</b> or <b>šké žávďarýí offskl</b> osti, afg <b>zstik[Lúklúd]</b> hím/pracovním souřadném
	↔ posun V Soyištádniem²-systému nástroje
12	* @param [in] offset_pos Posunutí pózy
13	* @param [in] spiral_param Parametr spirály
14	* @return Kód chyby
15	*/
16	errno_t NewSpiral(JointPos *joint_pos, DescPose *desc_pos, int tool, int user, float_
	, <b>float</b> acc, ExaxisPos *epos, <b>float</b> ovl, <b>uint8_t</b> offset_flag, DescPose *offset_pos,

### 2.1.3.11 Příklad kódu

```
#include <cstdlib>
1
   #include <iostream>
2
   #include <stdio.h>
3
   #include <cstring>
4
   #include <unistd.h>
5
   #include "MRobot.h"
6
   #include "RobotTypes.h"
7
8
   using namespace std;
9
10
   int main(void)
11
   {
12
        Robot MRobot;
                                            //Instituce objektu robota
13
        robot.RPC("192.168.58.2");
                                            /Navázání komunikačního spojení s robotem
14
      →kontrolér
15
        JointPos j;
16
        DescPose desc_pos, offset_pos1, offset_pos2;
17
        ExaxisPos epos;
18
        SpiralParam sp;
19
20
        memset(&j, 0, sizeof(JointPos));
21
        memset(&desc_pos, 0, sizeof(DescPose));
22
        memset(&offset_pos1, 0, sizeof(DescPose));
23
        memset(&offset_pos2, 0, sizeof(DescPose));
24
        memset(&epos, 0, sizeof(ExaxisPos));
25
        memset(&sp, 0, sizeof(SpiralParam));
26
27
        j = \{127.888, -101.535, -94.860, 17.836, 96.931, -61.325\};
28
        offset_pos1.tran.x = 50.0;
29
        offset_pos1.rpy.rx = -30.0;
30
        offset_pos2.tran.x = 50.0;
31
        offset_pos2.rpy.rx = -5.0;
32
33
        sp.circle_num = 5;
34
        sp.circle_angle = 5.0;
35
        sp.rad_init = 50.0;
36
        sp.rad_add = 10.0;
37
        sp.rotaxis_add = 10.0;
38
        sp.rot_direction = 0;
39
40
        int tool = 0;
41
        int user = 0;
42
        float vel = 100.0;
43
        float acc = 100.0;
44
        float ovl = 100,0;
45
        float blendT = 0.0;
46
        uint8_t flag = 2;
47
48
        robot.SetSpeed(20);
49
50
                                                                                       (pokračování na další straně)
```

```
int ret = robot.GetForwardKin(&j, &desc_pos); //Přední kinematické rozhraní může
51
      →použít k řešení kartézských souřadnic pouze s polohami kloubů
52
        if(ret == 0)
53
        £
54
            int err1 = robot.MoveJ(&j, &desc_pos, tool, user, vel, acc, ovl, &epos, blendT,
55
    , &offset_pos1);
            printf("movej errcode:%d\n", err1);
56
57
            int err2 = robot.NewSpiral(&j, &desc_pos, tool, user, vel, acc, &epos, ovl, flag,
58
    \rightarrow & offset_pos2, sp);
            printf("newspiral errcode:%d\n", err2);
59
60
        jinak
61
        Ł
62
            printf("GetForwardKin errcode:%d\n", ret);
63
64
        }
65
        vrátit 0;
66
   }
67
```

#### 2.1.3.12 Pohyb v kloubním prostoru v režimu serva

```
/**
1
   * @brief Pohyb v kloubním prostoru v režimu serva
2
   * @param [in] joint pos Poloha cílového kloubu, jednotka: deg
3
    * \widehat{@}param [in] acc Procentuální rozsah zrychlení[\widehat{0} \sim 100], zatím neotevřeno, výchozí: 0
4
5
   * @param [in] vel
                         Hodnota se pohybuje v rozmezí M om 0 až 100. Hodnota není k
   * dishamicjin hemdT Doba dodání pokynu, jednotka: s, doporučený rozsah [0.001~0.
6
    \rightarrow 0016 hodnota je 0
    * <sup>@param</sup> [in] filterT Doba filtrování (jednotka: s), dočasně vypnuto. Výchozí hodnota...
7
    \leftrightarrow je 0
                          Proporcionální zesilovač v cílové poloze, ještě není otevřený,
   * @param [v] zisk
8
    \rightarrow výchozí hodnota je 0
   * @return Kód chyby
9
10
   errno_t ServoJ(JointPos *joint_pos, float acc, float vel, float cmdT, float filterT,__
11
```

#### 2.1.3.13 Příklad kódu

```
1 #include <cstdlib>
```

```
2 #include <iostream>
```

- <sup>3</sup> *#include <stdio.h>*
- 4 *#include <cstring>*
- *s* #*include* <*unistd.h*>
- 6 #include "MRobot.h"

```
1 #include "RobotTypes.h"
```

```
8
9
   using namespace std;
10
   int main(void)
11
   Ł
12
                                            //Instituce objektu robota
        MRobot robot:
13
        robot.RPC("192.168.58.2");
                                            /Navázat komunikační spojení s robotem
14
    ⊶kontrolér
15
16
        JointPos j;
17
18
        memset(&j, 0, sizeof(JointPos));
19
        float vel = 0.0;
20
        float acc = 0.0;
21
        float cmdT = 0.008;
22
        float filterT = 0.0;
23
        float gain = 0.0;
24
        uint8_t flag = 0; int
25
        count = 500; float
26
        dt = 0.1;
27
28
        int ret = robot.GetActualJointPosDegree(flag, &j);
29
        if(ret == 0)
30
        {
31
             while (count)
32
             Ł
33
                 robot.ServoJ(&j, acc, vel, cmdT, filterT, gain);
34
                 j.jPos[0] += dt;
35
                 count -= 1;
36
                 robot.WaitMs(cmdT*1000);
37
             }
38
        jinak
39
40
        {
41
             printf("GetActualJointPosDegree errcode:%d\n", ret);
42
        }
43
44
        vrátit 0;
45
   }
46
```

### 2.1.3.14 Pohyb v režimu serva v kartézském prostoru

- <sup>6</sup> \* @param [in] acc Procentuální rozsah zrychlení[0~100], zatím neotevřeno, výchozí: 0
- \* @param [in] vel Hodnota se pohybuje v rozmezí M om 0 až 100. Hodnota není k diamozici. Tho
- <sup>8</sup> dispozici. The <sup>9</sup> di
- <sup>9</sup> \* @param [in] filterT Doba filtrování (jednotka: s), dočasně vypnuto. Výchozí hodnota → je 0

```
<sup>10</sup> * @param [v] zisk Proporcionální zesilovač v cílové poloze, ještě není otevřený, 

↔ výchozí hodnota je 0
```

```
11 * @return Kód chyby
```

```
12
13
```

```
errno_t ServoCart(int mode, DescPose *desc_pose, float pos_gain[6], float acc, float

, , float cmdT, float filterT, float gain);
```

### 2.1.3.15 Příklad kódu

```
#include <cstdlib>
1
   #include <iostream>
2
   #include <stdio.h>
3
   #include <cstring>
4
   #include <unistd.h>
5
   #include "MRobot.h"
6
   #include "RobotTypes.h"
7
8
   using namespace std;
9
10
   int main(void)
11
   £
12
        MRobot robot;
                                           //Instituce objektu robota
13
        robot.RPC("192.168.58.2");
                                           /Navázat komunikační spojení s robotem
14
    ⊶kontrolér
15
        DescPose desc_pos_dt;
16
        memset(&desc_pos_dt, 0, sizeof(DescPose));
17
18
        desc_pos_dt.tran.z = -0,5;
19
        float pos_gain[6] = {0.0,0.0,1.0,0.0,0.0,0.0,0.0};
20
        int mode = 2: float
21
        vel = 0.0; float acc
22
        = 0.0;
23
        float cmdT = 0,008;
24
        float filterT = 0,0;
25
        float gain = 0,0;
26
        uint8_t flag = 0; int
27
        count = 100;
28
29
        robot.SetSpeed(20);
30
31
        while (count)
32
        {
33
                                                                                      (pokračování na další straně)
```

34

35

36

37 38

39

40

(pokračování na předchozí straně)

```
robot.ServoCart(mode, &desc_pos_dt, pos_gain, acc, vel, cmdT, filterT, gain);
count -= 1;
robot.WaitMs(cmdT*1000);
}
vrátit 0;
}
```

#### 2.1.3.16 Pohyb z bodu do bodu v kartézském prostoru

```
/**
1
   * @brief Pohyb z bodu do bodu v kartézském prostoru
2
   * (a) param [v]
                    desc pos Cílová kartézská pozice nebo přírůstek
3
   * @param fight@ol Číslo souřadnice nástroje, rozsah [1~15] Číslo
   * @param [in] user souřadnice obrobku, rozsah [1~15] Procento
5
   * @param [in] vel
6
                        otáček, rozsah [0~100]
   * @param [in] acc
7
                        Procento zrychlení, rozsah [0~100], zatím neotevřeno
   * @param [in] ovl
8
   * @param [in] blendFalten škáhování mesterstil kovahlo (1609b0.01- doba vyhlazování)
9
   →(neblokující), v ms
10
   * @param [in] config
                           Konfigurace kloubního prostoru, [-1]- odkaz na aktuální
     kloub,
11
   \rightarrowpozice, [0~7]- vztahuje se ke konkrétní konfiguraci kloubního prostoru, výchozí
12
   hodnota je -1
13
   * @return Kód chyby
   */
            MoveCart(DescPose *desc_pos, int tool, int user, float vel, float acc, float
   errno_t
    , float blendT, int config);
   2.1.3.17 Příklad kódu
   #include <cstdlib>
1
   #include <iostream>
2
   #include <stdio.h>
3
   #include <cstring>
4
   #include <unistd.h>
5
   #include "MRobot.h"
6
   #include "RobotTypes.h"
7
8
   using namespace std;
9
10
   int main(void)
11
   ł
12
                                        //Instituce objektu robota
       MRobot robot;
13
```

```
<sup>14</sup> robot.RPC("192.168.58.2"); /Navázat komunikační spojení s robotem...

↔ kontrolér
```

```
<sup>15</sup>
16 DescPose desc_pos1, desc_pos2, desc_pos3;
```

```
memset(&desc_pos1, 0, sizeof(DescPose));
```

```
17 memset(&desc_pos1, 0, sizeof(DescPose))
18 memset(&desc_pos2, 0, sizeof(DescPose));
```

```
<sup>19</sup> memset(&desc_pos3, 0, sizeof(DescPose));
```

20	
21	desc_pos1.tran.x = 75.414;
22	desc_pos1.tran.y = 568.526;
23	desc_pos1.tran.z = 338.135;
24	desc_pos1.rpy.rx = -178.348;
25	$desc_pos1.rpy.ry = -0.930;$
26	desc. $pos1.rpv.rz = 52.611$ :
27	
28	desc_pos2.tran.x = -273.856;
29	desc_pos2.tran.y = 643.260;
30	desc_pos2.tran.z = 259.235;
31	desc_pos2.rpy.rx = -177.972;
32	desc_pos2.rpy.ry = -1.494;
33	desc pos $2$ .rpv.rz = 80.866:
34	
35	desc_pos3.tran.x = -423.044;
36	desc_pos3.tran.y = 229.703;
37	$desc_{pos}3.tran.z = 241.080;$
38	desc_pos3.rpy.rx = -173.990;
39	desc_pos3.rpy.ry = -5.772;
40	desc_pos3.rpy.rz = 123.971;
41	
42	int tool = 0;
43	<pre>int user = 0; float</pre>
44	vel = 100.0; float
45	acc = 100.0; tloat
46	OVI = 100.0;
47	$\frac{10at}{1} \text{ blend} = -1.0;$
48	int config = 1:
49	Int comig – -1,
50	robot.SetSpeed(20):
51	robot MoveCart(&desc post tool user vel acc ovl blendT config).
52	robot.MoveCart(&desc_post, tool, user, vel, acc. ovl, blendT, config);
53	robot.MoveCart(&desc_pos3, tool, user, vel, acc, ovl, blendT1, config):
54	,,,,,,,
55	vrátit O;
56	
57	}
	·

### 2.1.3.18 Začíná pohyb spline

```
1 /**
2 * @brief Začíná pohyb spline
3 * @return Kód chyby
4 */
5 errno_t_SplineStart();
```

### 2.1.3.19 Drážkový pohyb PTP

```
/**
1
   * @brief Pohyb kloubního prostoru
2
   * @param [in] joint pos Poloha cílového kloubu,
3
   * @param [indaka:pasg Cílová kartézská poloha
4
   * @param [in] tool Číslo souřadnice nástroje, rozsah [1~15] Číslo
5
   * @param [in] user souřadnice obrobku, rozsah [1~15] Procento
6
   * @param [in] vel
7
                        otáček, rozsah [0~100]
   * @param [in] acc
8
                       Procento zrychlení, rozsah [0~100], zatím neotevřeno
   * @param [in] ovl
9
   * @return Kód chybyFaktor škálování rychlosti, rozsah[0~100]
10
   */
11
   errno_t SplinePTP(JointPos *joint_pos, DescPose *desc_pos, int tool, int user, float_
12
   →vel, float acc, float ovl);
```

### 2.1.3.20 Pohyb drážkování končí

```
1 /**
2 * @brief Pohyb spline je dokončen
3 * @return Kód chyby
4 */
5 errno_t_SplineEnd();
```

### 2.1.3.21 Příklad kódu

```
#include <cstdlib>
1
   #include <iostream>
2
   #include <stdio.h>
3
   #include <cstring>
4
   #include <unistd.h>
5
   #include "MRobot.h"
6
   #include "RobotTypes.h"
7
8
   using namespace std;
9
10
   int main(void)
11
   Ł
12
                                           //Instituce objektu robota
        MRobot robot:
13
        robot.RPC("192.168.58.2");
                                           /Navázat komunikační spojení s robotem
14
    ↔kontrolér
15
        JointPos j1, j2, j3, j4;
16
        DescPose desc_pos1,desc_pos2,desc_pos3,desc_pos4,offset_pos;
17
        ExaxisPos epos;
18
19
        memset(&j1, 0, sizeof(JointPos));
20
        memset(&j2, 0, sizeof(JointPos));
21
        memset(&j3, 0, sizeof(JointPos));
22
        memset(&j4, 0, sizeof(JointPos));
23
                                                                                      (pokračování na další straně)
```

```
(pokračování na předchozí straně)
```

```
memset(&desc_pos1, 0, sizeof(DescPose));
24
        memset(&desc_pos2, 0, sizeof(DescPose));
25
        memset(&desc_pos3, 0, sizeof(DescPose));
26
        memset(&desc_pos4, 0, sizeof(DescPose));
27
        memset(&offset_pos, 0, sizeof(DescPose));
28
        memset(&epos, 0, sizeof(ExaxisPos));
29
30
        j1 = \{114.578, -117.798, -97.745, -54.436, 90.053, -45.216\};
31
        desc_{pos1.tran.x} = -140.418;
32
        desc_pos1.tran.y = 619.351;
33
        desc_pos1.tran.z = 198.369;
34
        desc_pos1.rpy.rx = -179.948;
35
        desc_pos1.rpy.ry = 0.023;
36
        desc_pos1.rpy.rz = 69.793;
37
38
        j2 = \{115.401, -105.206, -117.959, -49.727, 90.054, -45.222\};
39
        desc_pos2.tran.x = -95.586;
40
        desc_pos2.tran.y = 504.143;
41
        desc_pos2.tran.z = 186.880;
42
        desc_pos2.rpy.rx = 178.001;
43
        desc_pos2.rpy.ry = 2.091;
44
        desc_pos2.rpy.rz = 70.585;
45
46
        j3 = \{135.609, -103.249, -120.211, -49.715, 90.058, -45.219\};
47
        desc_pos3.tran.x = -252.429;
48
        desc_pos3.tran.y = 428.903;
49
        desc_pos3.tran.z = 188.492;
50
        desc_pos3.rpy.rx = 177.804;
51
        desc_pos3.rpy.ry = 2.294;
52
        desc_pos3.rpy.rz = 90.782;
53
54
        j4 = \{154.766, -87.036, -135.672, -49.045, 90.739, -45.223\};
55
        desc_pos4.tran.x = -277.255;
56
        desc_pos4.tran.y = 272.958;
57
        desc_pos4.tran.z = 205.452;
58
        desc_pos4.rpy.rx = 179.289;
59
        desc_pos4.rpy.ry = 1.765;
60
        desc_pos4.rpy.rz = 109.966;
61
62
        int tool = 0;
63
        int user = 0;
64
        float vel = 100.0;
65
        float acc = 100.0;
66
        float ovl = 100.0;
67
        float blendT = -1.0;
68
        uint8_t flag = 0;
69
70
        robot.SetSpeed(20);
71
72
        int err1 = robot.MoveJ(&j1, &desc_pos1, tool, user, vel, acc, ovl, &epos, blendT,
73
    , &offset_pos);
        printf("movej errcode:%d\n", err1);
74
```

```
(pokračování na předchozí straně)
```

```
robot.SplineStart();
75
       robot.SplinePTP(&j1, &desc_pos1, tool, user, vel, acc, ovl);
76
       robot.SplinePTP(&j2, &desc_pos2, tool, user, vel, acc, ovl);
77
       robot.SplinePTP(&j3, &desc_pos3, tool, user, vel, acc, ovl);
78
       robot.SplinePTP(&j4, &desc_pos4, tool, user, vel, acc, ovl);
79
       robot.SplineEnd();
80
81
       vrátit 0;
82
   }
83
```

### 2.1.3.22 Návrh na ukončení

```
1 /**
2 * @brief Návrh na ukončení
3 * @return Kód chyby
4 */
5 errno_t_StopMotion();
```

### 2.1.3.23 Celý bodový posun začíná

1 /\*\*
 2 \* @brief Celý bodový posun začíná
 3 \* @param [in] flag 0- posun v základním souřadnicovém systému/souřadnicovém systému
 obrobku, 2-\_\_\_
 4 ↔ posun v souřadném systému nástroje
 5 \* @param [in] offset\_pos Posunutí pózy
 6 7 \* @return Kód chyby
 \*/

errno\_t PointsOffsetEnable(int flag, DescPose \*offset\_pos);

### 2.1.3.24 Celý bodový posun končí

```
1 /**
2 * @brief Celý bodový posun končí
3 * @return Kód chyby
4 */
5 errno t PointsOffsetDisable();
```

### 2.1.3.25 Příklad kódu

```
<sup>1</sup> #include <cstdlib>
```

```
2 #include <iostream>
```

```
<sup>3</sup> #include <stdio.h>
```

```
4 #include <cstring>
```

```
5 #include <unistd.h>
```

```
6 #include "MRobot.h"
7 #include "RobotTypes.h"
```
```
using namespace std;
9
10
   int main(void)
11
   {
12
        Robot MRobot;
                                            //Instituce objektu robota
13
        robot.RPC("192.168.58.2");
                                            /Navázání komunikačního spojení s robotem
14
      →kontrolér
    15
        JointPos j1,j2;
16
        DescPose desc_pos1,desc_pos2,offset_pos,offset_pos1;
17
        ExaxisPos epos;
18
19
        memset(&j1, 0, sizeof(JointPos));
20
        memset(&j2, 0, sizeof(JointPos));
21
        memset(&desc_pos1, 0, sizeof(DescPose));
22
        memset(&desc_pos2, 0, sizeof(DescPose));
23
        memset(&offset_pos, 0, sizeof(DescPose));
24
        memset(&offset_pos1, 0, sizeof(DescPose));
25
        memset(&epos, 0, sizeof(ExaxisPos));
26
27
        j1 = \{114.578, -117.798, -97.745, -54.436, 90.053, -45.216\};
28
        desc_{pos1.tran.x} = -140.418;
29
        desc_pos1.tran.y = 619.351;
30
        desc_pos1.tran.z = 198.369;
31
        desc_pos1.rpy.rx = -179.948;
32
        desc_pos1.rpy.ry = 0.023;
33
        desc_pos1.rpy.rz = 69.793;
34
35
        j2 = \{115.401, -105.206, -117.959, -49.727, 90.054, -45.222\};
36
        desc_pos2.tran.x = -95.586;
37
        desc_{pos2.tran.y} = 504.143;
38
        desc_pos2.tran.z = 186.880;
39
        desc_pos2.rpy.rx = 178.001;
40
        desc_pos2.rpy.ry = 2.091;
41
        desc_pos2.rpy.rz = 70.585;
42
43
        offset_pos1.tran.x = 100.0;
44
        offset_pos1.tran.y = 100.0;
45
        offset_pos1.tran.z = 100.0;
46
        offset_pos1.rpy.rx = 5.0;
47
        offset_pos1.rpy.ry = 5.0;
48
        offset_pos1.rpy.rz = 5.0;
49
50
        int tool = 0;
51
        int user = 0;
52
        float vel = 100.0;
53
        float acc = 100,0;
54
        float ovl = 100,0;
55
        float blendT = -1.0;
56
        float blendR = 0.0;
57
        uint8_t flag = 0;
58
                                                                                       (pokračování na další straně)
```

8

```
(pokračování na předchozí straně)
```

```
59
        int type = 0;
60
        robot.SetSpeed(20);
61
62
        robot.MoveJ(&j1, &desc_pos1, tool, user, vel, acc, ovl, &epos, blendT,flag, &offset_
63
    →pos);
        robot.MoveJ(&j2, &desc_pos2, tool, user, vel, acc, ovl, &epos, blendT,flag, &offset_
64
    <sub>↔pos</sub>);
        sleep(2);
65
        robot.PointsOffsetEnable(type, &offset_pos1);
66
        robot.MoveJ(&j1, &desc_pos1, tool, user, vel, acc, ovl, &epos, blendT,flag, &offset_
67
    , →pos);
        robot.MoveJ(&j2, &desc_pos2, tool, user, vel, acc, ovl, &epos, blendT,flag, &offset_
68
    , →pos);
        robot.PointsOffsetDisable();
69
70
        vrátit 0:
71
   }
72
```

# 2.1.4 IO

# 2.1.4.1 Nastavení digitálního výstupu řídicí jednotky

```
/**
1
   * @brief Nastavení digitálního výstupu řídicí jednotky
2
3
   * @param [in] id I/O číslo a rozsah[0~15]
4
   * @param [in] status 0- off, 1- on
5
   * @param [in] smooth 0- Není hladký, 1- hladký
6
                          0- blocking, 1- non-blocking
   * @param [in] blok
7
   * @return Kód chyby
8
   */
9
```

<sup>L</sup>errno\_t <u>SetDO(int id, uint8\_t status, uint8\_t smooth, uint8\_t block);</u>

# 2.1.4.2 Nastavení digitálního výstupu nástroje

```
/**
1
   * @brief Nastavení digitálního výstupu nástroje
2
3
   * @param [in] id
                     I/O číslo a rozsah[0~1]
4
   * @param [in] status 0- off, 1- on
5
   * @param [in] smooth 0- není hladký, 1- hladký
6
   * @param [in] blok 0- blocking, 1- non-blocking
7
   * @return Kód chyby
8
   */
9
  errno_t <u>SetToolDO(int id, uint8 t status, uint8 t smooth, uint8 t block);</u>
```

#### 2.1.4.3 Nastavení analogového výstupu řídicí jednotky

1	/**
2	* @brief Nastavení analogového výstupu řídicí jednotky
3	* @param [in] id I/O číslo a rozsah[0~1]
4	* @param [in] hodnota Procento hodnoty proudu nebo napětí, rozsah [0~100] odpovídající
	↔na hodnotu proudu [0~20mA] nebo napětí [0~10V]
5	* @param [in] blok 0- blocking, 1- non-blocking
6	* @return Kód chyby
7	*/
8	errno_t_SetAO(int_id, float_value, uint8_t_block);



```
/**
1
   * @brief Nastavení analogového výstupu nástroje
2
   * @param [in] id I/O číslo, rozsah [0]
3
4
   * @param [in] hodnota Procento hodnoty proudu nebo napětí, rozsah [0~100] odpovídající
   ↔na hodnotu proudu [0~20mA] nebo napětí [0~10V]
5
   * @param [in] blok
                       0- blocking, 1- non-blocking
6
   * @return Kód chyby
   */
7
8
  <u>errno_t_SetToolAO(int_id, float_value, uint8_t_block);</u>
```

# 2.1.4.5 Získejte digitální vstup ovládací skříňky



#### 2.1.4.6 Získat číselný vstup nástroje

```
/**
1
   * @brief Získat číselný vstup nástroje
2
3
   * @param [in] id I/O číslo, rozsah[0~1]
4
   * @param [in] blok 0- blocking, 1- non-blocking
5
   * @param [out] výsledek 0- low, 1- high
6
   * @return Kód chyby
7
   */
8
  errno_t GetToolDI(int id, uint8 t block, uint8 t *result);
```

#### 2.1.4.7 Počkejte na digitální vstup řídicí jednotky

1	/**
2	* @brief Počkejte na digitální vstup řídicí
3	<i>fedeportup</i> [in] id I/O numberrange[0~15]
4	* @param [v] stav 0- vypnuto, 1- zapnuto
5	* @param [v] max_time Maximální čekací doba vyjádřená v ms
6	* @param [v] opt Po vypršení časového limitu, 0- program se zastaví a vyzve k vypršení
	$ \rightarrow ignoruje výzvy k vyprsení časového limitu a pokračuje v provádění, 2-$
7	čeká.
8	* @return Kód chyby
9	*/
	errno_t WaitDI(int id, uint8_t status, int max_time, int opt);

#### 2.1.4.8 Čekání na digitální vstup multiplexu řídicí jednotky

1	/**
2	* @brief Čekání na digitální vstup multiplexu řídicí jednotky
3	* @param [in] režim 0- multiplexovaný a, 1- multiplexovaný nebo
4	* @param [in] id I/O čísla. bit0 až bit7 odpovídá DI0 až DI7 a bit8 a <b>ž</b>
5	* @param [in] status 0- off 1- on
6	* @param [in] max_time Maximální čekací doba vyjádřená v ms
7	* @param [in] opt Po uplynutí politiky časového limitu, 0- program se zastaví a nabídne časový
9	limit, 1
	↔ Ignoruje výzvy k vypršení časového limitu a pokračuje v provádění, 2- čeká.
9	* @return Kód chyby
10	*/

errno\_t WaitMultiDI(int mode, int id, uint8\_t status, int max\_time, int opt);

#### 2.1.4.9 Počkejte na zadání čísla nástroje

```
/**
1
   * @brief Počkejte na zadání čísla nástroje
2
   * @param [in] id I/O numbersrange[0~1]
3
   * @param [v] stav 0- vypnuto, 1- zapnuto
* @param [v] max_time Maximální čekací doba vyjádřená v ms
4
5
   * @param [v] opt Po vypršení časového limitu, 0- program se zastaví a vyzve k vypršení
6
   ·→ ignoruje výzvy k vypršenúčasového_limitu a pokračuje v provádění, 2-
   čeká.
7
   * @return Kód chyby
8
   */
9
```

```
errno_t WaitToolDI(int id, uint8_t status, int max_time, int opt);
```

### 2.1.4.10 Získat analogový vstup řídicí jednotky

1	/**
2	* @brief Získání analogového vstupu ovládací skříňky
3	* @param [in] id I/O numbersrange[0~1]
4	* @param [in] blok 0- blocking, 1- non-blocking
5	* @param [out] výsledek Procento hodnoty vstupního proudu nebo napětí, rozsah [0-100] _ ↔odpovídající hodnotě proudu [0-20ms] nebo napětí [0-10V]
6	* @return Kód chyby
7	*/
8	errno_t GetAI(int id, uint8_t block, float *result);



```
/**
1
  * @brief Získat analogový vstup nástroje
2
  * @param [in] id I/O numbersrange[0~1]
3
  * @param [in] blok
                        0- blocking, 1- non-blocking
4
5
  * @param [out] výsledek Procento hodnoty vstupního proudu nebo napětí, rozsah [0-100]
   ↔ odpovídající hodnotě proudu [0-20ms] nebo napětí [0-10V]
6
  * @return Kód chyby
   */
7
8
  errno_t_GetToolAI(int_id, uint8_t_block, float *result);
```

# 2.1.4.12 Čekání na analogový vstup řídicí jednotky

1	/**
2	* @brief Čekání na analogový vstup řídicí jednotky
3	* @param [in] id I/O numbersrange[0~1]
4	* @param [in] znaménko 0-větší než1-menší než
5	* @param [in] value Procento hodnoty vstupního proudu nebo napětí, rozsah [0-100] →odpovídající hodnotě proudu [0-20ms] nebo napětí [0-10V]
6	* (aparam [in] max_time Maximalni Cekaci doba vyjadrena v ms
7	* @param [in] opt Po uplynutí politiky časového limitu, 0- program se zastaví a nabídne časový
8	limit, 1
9	↔ <sub>ignoruje</sub> výzvy k vypršení časového limitu a pokračuje v provádění, 2- čeká.
10	* @return Kód chyby */

errno\_t WaitAI(int id, int sign, float value, int max\_time, int opt);

# 2.1.4.13 Čekání na analogový vstup nástroje

1	/**
2	* @brief Čekání na analogový vstup nástroje
3	* @param [in] id I/O numbersrange[0~1]
4	* @param [in] znaménko 0-větší než1-menší než
5	* @param [in] value Procento hodnoty vstupního proudu nebo napětí, rozsah [0-100]
	↔odpovídající hodnotě proudu [0-20ms] nebo napětí [0-10V]
6	* @param [in] max_time Maximální čekací doba vyjádřená v ms
	(pokračování na další straně)

```
    * @param [in] opt Po uplynutí politiky časového limitu, 0- program se zastaví a nabídne časový limit, 1-___
    ↔ Ignoruje výzvy k vypršení časového limitu a pokračuje v provádění, 2- čeká.
    * @return Kód chyby
    */
```

errno\_t WaitToolAI(int id, int sign, float value, int max\_time, int opt);

#### 2.1.4.14 Příklad kódu

```
#include <cstdlib>
1
   #include <iostream>
2
   #include <stdio.h>
3
   #include <cstring>
4
   #include <unistd.h>
5
   #include "MRobot.h"
6
   #include "RobotTypes.h"
7
8
   using namespace std;
9
10
   int main(void)
11
   ł
12
                                           //Instituce objektu robota
        MRobot robot:
13
        robot.RPC("192.168.58.2");
                                           /Navázat komunikační spojení s robotem
14
    ↔kontrolér
15
        uint8_t status = 1;
16
        uint8_t smooth = 0;
17
        uint8 t block = 0;
18
        uint8_t di = 0, tool_di = 0; float
19
        ai = 0.0, tool_ai = 0.0; float value
20
        = 0.0:
21
        int i;
22
23
        for(i = 0; i < 16; i++)
24
        Ł
25
             robot.SetDO(i, status, smooth, block);
26
            robot.WaitMs(1000);
27
        }
28
29
        status = 0;
30
31
        for(i = 0; i < 16; i++)
32
        Ł
33
             robot.SetDO(i, status, smooth, block);
34
             robot.WaitMs(1000);
35
        }
36
37
        status = 1;
38
        for(i = 0; i < 2; i++)
39
        {
40
41
```

(pokračování na další straně)

```
(pokračování na předchozí straně)
```

```
robot.SetToolDO(i, status, smooth, block);
42
            robot.WaitMs(1000);
43
        }
44
45
        status = 0;
46
47
        for(i = 0; i < 2; i++)
48
        Ł
49
            robot.SetToolDO(i, status, smooth, block);
50
            robot.WaitMs(1000);
51
        }
52
53
        hodnota = 50,0;
54
        robot.SetAO(0, hodnota, blok);
55
        hodnota = 100,0;
56
        robot.SetAO(1, hodnota, blok);
57
        robot.WaitMs(1000);
58
        hodnota = 0,0;
59
        robot.SetAO(0, hodnota, blok);
60
        hodnota = 0,0;
61
        robot.SetAO(1, hodnota, blok);
62
63
        hodnota = 100.0;
64
        robot.SetToolAO(0, hodnota, blok);
65
        robot.WaitMs(1000);
66
        hodnota = 0,0;
67
        robot.SetToolAO(0, hodnota, blok);
68
69
        robot.GetDI(0, blok, &di);
70
        printf("di0:%u\n", di);
71
                                                 //Čekali jsme
        robot.WaitDI(0,1,0,2);
72
        robot.WaitMultiDI(1,3,3,10000,2);
                                                //Čekali jsme
73
        tool_di = robot.GetToolDI(1, blok, &tool_di);
74
        printf("tool_di1:%u\n", tool_di);
75
        robot.WaitToolDI(1,1,0,2);
                                                 //Čekali jsme
76
77
        robot.GetAI(0,block, &ai);
78
        printf("ai0:%f\n", ai);
79
        robot.WaitAI(0,0,50,0,2);
                                                 //Čekali jsme
80
        robot.WaitToolAI(0,0,50,0,2);
                                                //Čekali jsme
81
        tool_ai = robot.GetToolAI(0,block, &tool_ai);
82
        printf("tool_ai0:%f\n", tool_ai);
83
84
        návrat 0;
85
   }
86
```

# 2.1.5 Společná nastavení

#### 2.1.5.1 Nastavení globální rychlosti

```
/**
* @brief Nastavení globální rychlosti
* @param [in] vel Procento rychlosti, rozsah[0~100]
* @return Kód chyby
*/
```

lerrno\_t\_SetSpeed(int\_vel);

#### 2.1.5.2 Nastavení hodnoty systémové proměnné

1 /\*\*
2 \* @brief Nastavení hodnoty systémové proměnné
3 \* @param [in] id Číslo proměnné, rozsah[1~20]
4 \* @param [in] value Hodnota proměnné
5 \* @return Kód chyby
\*/
7

errno\_t <u>SetSysVarValue(int id, float value);</u>

#### 2.1.5.3 Nastavení souřadnicového systému nástroje

/\*\* 1 \* @brief Nastavení souřadnicového systému nástroje 2 \* @param [in] id M ame number, range[1~15] 3 4 \* @param [in] coord Poloha středu nástroje vzhledem k poloze středu koncové příruby 5 \* @param [in] type 0- souřadnice nástroje, 1- souřadnice senzoru 6 \* @param [in] install Pozice instalace, 0- konec robota, 1- vně robota 7 \* @return Kód chvbv 8 9

errno\_t SetToolCoord(int id, DescPose \*coord, int type, int install);

#### 2.1.5.4 Nastavení seznamu souřadnic nástroje

/\*\* 1 \* @brief Nastavení seznamu souřadnic nástroje 2 3 \* @param [in] id M ame number, range[1~15] 4 \* @param [in] coord Poloha středu nástroje vzhledem k poloze středu koncové příruby \* @param [in] type 0- souřadnice nástroje, 1- souřadnice senzoru 6 \* @param [in] install Pozice instalace, 0- konec robota, 1- vně robota 7 \* @*return* Kód chyby 8 \*/ 9 errno\_t SetToolList(int id, DescPose \*coord, int type, int install);

#### 2.1.5.5 Nastavení externího souřadnicového systému nástroje

l	/**
2	* @brief Nastavení externího souřadnicového systému nástroje
3	* @param [in] id M ame number, range[1~15]
1	* @param [in] etcp Poloha středu nástroje vzhledem k poloze středu koncové příruby
5	* @param [in] etool Určuje se
5	* @return Kód chyby
2	*/
,	errno_t SetExToolCoord(int id, DescPose *etcp, DescPose *etool);

#### 2.1.5.6 Nastavení seznamu externích souřadnicových systémů nástroje

```
/**
1
   * @brief Nastavení seznamu externích souřadnicových systémů nástroje
2
3
   * (a)param [in] id M ame number, range[1~15]
4
  * @param [in] etcp Poloha středu nástroje vzhledem k poloze středu koncové příruby
5
  * @param [in] etool Určuje se
6
   * @return Kód chyby
7
   */
8
```

errno\_t SetExToolList(int id, DescPose \*etcp, DescPose \*etcol);

#### 2.1.5.7 Nastavení souřadnicového systému obrobku

/\*\* 1 \* @brief Nastavení souřadnicového systému obrobku 2 3 \* @param [in] id M ame number, range[1~15] \* @param [in] coord Poloha středu nástroje vzhledem k poloze středu koncové příruby 4 5 \* @*return* Kód chyby 6 \*/ errno\_t SetWObjCoord(int id, DescPose \*coord);

#### 2.1.5.8 Nastavení seznamu pracovních souřadnicových systémů

```
/**
1
  * @brief Nastavení seznamu pracovních souřadnicových systémů
2
3
  * @param [in] id M ame number, range[1~15]
4
  * @param [in] coord Poloha středu nástroje vzhledem k poloze středu koncové příruby
5
  * @return Kód chyby
  */
6
```

errno\_t SetWObjList(**int** id, DescPose \*coord);

### 2.1.5.9 Nastavení hmotnosti koncového zatížení

```
1 /**
2 * @brief Nastavení hmotnosti koncového zatížení
3 * @param [in] weight Hmotnost nákladu, jednotka: kg
4 * @return Kód chyby
5 */
6 errno_t SetLoadWeight(float weight);
```



1 /\*\*
2 \* @brief Nastavení souřadnic centroidu koncového zatížení
3 \* @param [in] coord Souřadnice centroidu, jednotka: mm
4 \* @return Kód chyby
5 \*/
6 errno\_t \_\_SetLoadCoord(DescTran \*coord);

# 2.1.5.11 Nastavení režimu instalace robota

```
/**

@brief Nastavení režimu instalace robota
@param [in] install Režim instalace: 0- formální instalace, 1- vedlejší instalace, ...
~>2- obrácená instalace

@return Kód chyby

*/

ferrno_t SetRobotInstallPos(uint8_t install);
```

#### 2.1.5.12 Nastavení instalace robota Úhel



#### 2.1.5.13 Počkejte na zadaný čas

```
1 /**
2 /**
2 * @brief Počkat na zadaný čas
3 * @param [in] t_ms jednotka: ms
4 * @return Kód chyby
5 */
6 errno_t_WaitMs(int_t_ms);
```

#### 2.1.5.14 Příklad kódu

```
#include <cstdlib>
1
   #include <iostream>
2
   #include <stdio.h>
3
   #include <cstring>
4
   #include <unistd.h>
5
   #include "MRobot.h"
6
   #include "RobotTypes.h"
7
8
   using namespace std;
9
10
   int main(void)
11
   {
12
        Robot MRobot;
                                            //Instituce objektu robota
13
        robot.RPC("192.168.58.2");
                                            /Navázání komunikačního spojení s robotem
14
    ↔kontrolér
15
        int i:
16
        hodnota float;
17
        int id;
18
        typ int;
19
        int install;
20
21
        DescTran coord;
22
        DescPose t_coord, etcp, etool, w_coord;
23
        memset(&coord, 0, sizeof(DescTran));
24
        memset(&t_coord, 0, sizeof(DescPose));
25
        memset(&etcp, 0, sizeof(DescPose));
26
        memset(&etool, 0, sizeof(DescPose));
27
        memset(&w_coord, 0, sizeof(DescPose));
28
29
        robot.SetSpeed(20);
30
31
        for(i = 1; i < 21; i++)
32
        Ł
33
             robot.SetSysVarValue(i, i+0.5);
34
             robot.WaitMs(1000);
35
        }
36
37
        for(i = 1; i < 21; i++)
38
        Ł
39
             robot.GetSysVarValue(i, &value);
40
             printf("sys value:%f\n", value);
41
        }
42
43
        robot.SetLoadWeight(2.5);
44
45
        coord.x = 3.0;
46
        coord.y = 4.0;
47
        coord.z = 5.0;
48
49
        robot.SetLoadCoord(&coord);
50
                                                                                       (pokračování na další straně)
```

(pokračování na předchozí straně)

51	
52	id = 10;
53	t_coord.tran.x = 1.0;
54	t_coord.tran.y = 2.0;
55	t_coord.tran.z = 3,0;
56	t_coord.rpy.rx = 4.0;
57	t_coord.rpy.ry = 5.0;
58	t_coord.rpy.rz = 6,0;
59	type = 0;
60	1 nstalovat = 0;
61	robot.SetToolCoord(id, &t_coord, type, install);
62 63	robot.SetToolList(id, &t_coord, type, install);
64	etcp.tran.x = 1.0;
65	etcp.tran.y = 2.0;
66	etcp.tran.z = 3.0;
67	etcp.rpy.rx = 4.0;
68	etcp.rpy.ry = 5.0;
69	etcp.rpy.rz = 6.0;
70	etool.tran.x = 11.0;
71	etool.tran.y = 22.0;
72	etool.tran.z = 33.0;
73	etool.rpy.rx = 44.0;
74	etool.rpy.ry = 55.0;
75	$e_{1001.1}p_{y.12} - 00.0;$
76	rabot SatExToolCoord(id Staten Stateol):
TI	
78 79	robot.SetExToolList(id, &etcp, &etool);
80	w_coord.tran.x = 11.0;
81	w_coord.tran.y = 12.0;
82	$w\_coord.tran.z = 13.0;$
83	w_coord.rpy.rx = 14.0;
84	w_coord.rpy.ry = 15.0;
85	w_coord.rpy.rz = 16,0;
86	id = 12;
87	robot.SetWObjCoord(id, &w_coord);
88 89	robot.SetWObjList(id, &w_coord);
90	robot.SetRobotInstallPos(0);
91 92	robot.SetRobotInstallAngle(15.0,25.0);
93	vrátit 0;
94	}

# 2.1.6 Nastavení zabezpečení

#### 2.1.6.1 Nastavení úrovně kolize

/\*\* 1 \* @brief Nastavení úrovně kolize 2 3 \* @param [in] mode 0- stupeň, 1- procento 4

- \* @param [in] level Collision threshold, grade range [], percentage range [0~1] 5 \* @param [in] config 0- Neaktualizovat konfigurační soubor. 1- Aktualizujte
- *↔konfigurační soubor*
- 6 \* @return Kód chyby

7 8

1

4

5

6

\*/

errno\_t SetAnticollision(int mode, float level[6], int config);

# 2.1.6.2 Nastavení zásad po kolizi

/\*\* 1 \* @brief Nastavení zásad po kolizi 2 3 \* @param [in] strategie 0- Chyba stop, 1- Pokračovat v běhu 4 \* @return Kód chyby 5 \*/ 6 errno\_t\_SetCollisionStrategy(int\_strategy);

## 2.1.6.3 Nastavení kladného limitu

/\*\* \* @brief Nastavení kladného limitu 2 3 \* @param [in] limit Šest pozic kloubů, jednotka: deg \* @return Kód chyby \*/ errno\_t\_SetLimitPositive(float\_limit[6]);

#### 2.1.6.4 Nastavení záporného limitu

/\*\* 1 \* @brief Nastavení záporného limitu 2 3 \* @param [in] limit Šest pozic kloubů, jednotka: deg 4 \* @return Kód chyby 5 \*/ 6 errno\_t\_SetLimitNegative(float\_limit[6]);

#### 2.1.6.5 Vymazání chybového stavu

```
1 /**
2 * @brief Vymazání chybového stavu
3 * @return Kód chyby
4 */
5 errno t ResetAllError();
```

#### 2.1.6.6 Společný přepínač kompenzace m iction

```
1 /**
2 * @brief Společný přepínač kompenzace m iction
3 * @param [in] state 0- vypnuto, 1- zapnuto
4 * @return Kód chyby
5 */
6 errno t MictionCompensationOnOff(uint8_t state);
```



1 /\*\*
2 @brief Nastavení společného koeficientu kompenzace M - formální
3 @param [in] coeff Šest kloubních kompenzačních koeficientů, rozsah [0~1]
4 @return Kód chyby
5 errno\_t SetMictionValue\_level(float coeff[6]);



```
1 /**
2 /**
2 @brief Nastavení koeficientu kompenzace kloubu M iction - boční montáž
3 @param [in] coeff Šest kloubních kompenzačních koeficientů, rozsah [0~1]
4 @return Kód chyby
5 errno_t SetMictionValue_wall(float coeff[6]);
```

#### 2.1.6.9 Nastavení koeficientu kompenzace kloubu M - zpětná montáž

1	/**
2	* @brief Nastavení společného koeficientu kompenzace M - obráceně
3	* @param [in] coeff Šest kloubových kompenzačních koeficientů, rozsah $[0~1]$
4	* @return Kód chyby
5	*/
6	errno_t_SetMictionValue_ceiling(float_coeff[6]);

#### 2.1.6.10 Nastavení společného koeficientu kompenzace M iction - M ee mount

```
/**

* @brief Nastavení společného koeficientu kompenzace Miction - Mee mount

* @param [in] coeff Šest kloubních kompenzačních koeficientů, rozsah [0~1]

* @return Kód chyby

*/

errno_t SetMictionValue_Meedom(float coeff[6]);
```

#### 2.1.6.11 Příklad kódu

1

2

3 4

5

6

```
#include <cstdlib>
1
   #include <iostream>
2
   #include <stdio.h>
3
   #include <cstring>
4
   #include <unistd.h>
5
   #include " MRobot.h"
6
   #include "RobotTypes.h"
7
8
   using namespace std;
9
10
   int main(void)
11
   {
12
        Robot MRobot;
                                             //Instituce objektu robota
13
        robot.RPC("192.168.58.2");
                                             /Navázání komunikačního spojení s robotem
14
      →kontrolér
15
        int mode = 0;
16
        int config = 1;
17
        float level1[6] = \{1.0, 2.0, 3.0, 4.0, 5.0, 6.0\};
18
        float level2[6] = \{50.0, 20.0, 30.0, 40.0, 50.0, 60.0\};
19
20
        robot.SetAnticollision(mode, level1, config);
21
        mode = 1:
22
        robot.SetAnticollision(mode, level2, config);
23
        robot.SetCollisionStrategy(1);
24
25
        float plimit[6] = \{170.0, 80.0, 150.0, 80.0, 170.0, 160.0\};
26
        robot.SetLimitPositive(plimit);
27
        float nlimit[6] = \{-170.0, -260.0, -150.0, -260.0, -170.0, -160.0\};
28
        robot.SetLimitNegative(nlimit);
29
30
        robot.ResetAllError();
31
32
        float lcoeff[6] = \{0.9, 0.9, 0.9, 0.9, 0.9, 0.9, 0.9\};
33
        float wcoeff[6] = \{0.4, 0.4, 0.4, 0.4, 0.4, 0.4, 0.4\};
34
        float ccoeff[6] = {0.6,0.6,0.6,0.6,0.6,0.6,0.6};
35
        float fcoeff[6] = \{0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5\};
36
        robot.MictionCompensationOnOff(1);
37
        robot.SetMictionValue_level(lcoeff);
38
        robot.SetMictionValue_wall(wcoeff);
39
                                                                                         (pokračování na další straně)
```

# 2.1.7 Stavový dotaz

#### 2.1.7.1 Získání úhlu montáže robota

1 /\*\*
2 \* @brief Získání úhlu montáže robota
3 \* @param [out] yangle Úhel sklonu
4 \* @param [out] zangle Úhel oMotation
5 \* @return Kód chyby
7

errno\_t GetRobotInstallAngle(float \*yangle, float \*zangle);

# 2.1.7.2 Získání hodnoty systémové proměnné

1 /\*\*
2 /\*\*
2 @brief Získání hodnoty systémové proměnné
3 @param [in] id Číslo systémové proměnné, rozsah[1~20]
4 @param [out] hodnota Hodnota systémové proměnné
5 @return Kód chyby
6 /
7 errno\_t GetSysVarValue(int id, float \*value);



1 /\*\*
2 \* @brief Získání aktuální polohy kloubu (úhel)
3 \* @param [in] příznak 0- blocking, 1- non-blocking
4 \* @param [out] jPos Šest pozic kloubů, jednotka: deg
5 \* @return Kód chyby
6 \*/

errno\_t GetActualJointPosDegree(uint8\_t flag, JointPos \*jPos);

7

## 2.1.7.4 Získání aktuální polohy kloubu (v radiánech)

5

6

7

\* @brief Získání aktuální polohy kloubu (v radiánech)

\* @param [in] příznak 0- blocking, 1- non-blocking

\* @param [out] jPos Šest pozic kloubů, jednotka: rad

\* @return Kód chyby

errno\_t GetActualJointPosRadian(uint8\_t flag, JointPos \*jPos);

# 2.1.7.5 Získat aktuální pozici nástroje

1 /\*\*
2 \* @brief Získat aktuální pozici nástroje
3 \* @param [in] flag 0- blocking, 1- non-blocking
4 \* @param [out] desc\_pos Pozice nástroje
5 \* @return Kód chyby
6 \*/
7

errno\_t GetActualTCPPose(uint8\_t flag, DescPose \*desc\_pos);

# 2.1.7.6 Získání aktuálního čísla souřadného systému nástroje

-errno\_t \_\_GetActualTCPNum(<mark>uint8\_t\_flag, int\_</mark>\*id);\_\_

# 2.1.7.7 Získání aktuálního čísla souřadnicového systému obrobku

1 /\*\*
2 @brief Získání aktuálního čísla souřadnicového systému obrobku
3 @param [in] flag 0- blocking, 1- non-blocking
4 @param [out] id Číslo souřadnicového systému úlohy
5 @return Kód chyby
6 \*/
7 errno\_t GetActualWObjNum(uint8 t flag, int \*id);

#### 2.1.7.8 Získání aktuální polohy koncové příruby

 $\begin{array}{c}
1 \\
2 \\
3 \\
4 \\
5 \\
4
\end{array}$ 

6 7 \* @brief Získat aktuální polohu koncové příruby

\* @param [in] flag 0- blocking, 1- non-blocking

\* @param [out] desc\_pos Pozice příruby

\* @return Kód chyby

\*/

errno\_t \_\_GetActualToolFlangePose(<mark>uint8\_t\_flag, DescPose</mark> \*desc\_pos);

# 2.1.7.9 Řešení inverzní kinematiky

/\*\* 1 \* @brief Řešení inverzní kinematiky 2 3 \* @param [in] typ 0- absolutní pozice (základní M ame), 1- inkrementální pozice (základní M ame), 2-\_ *·→inkrementální pozice (nástroj M ame)* \* @param [in] desc pos Kartézská pozice 4 5 \* @param [in] config Konfigurace kloubního prostoru, [-1]- na základě aktuálního kloubu ↔poloha, [0~7]- na základě konkrétní konfigurace kloubního prostoru 6 \* @param [out] joint pos Poloha kloubu 7 \* @return Kód chyby \*/ 8 9 errno\_t GetInverseKin(int type, DescPose \*desc\_pos, int config, JointPos \*joint\_pos);

# 2.1.7.10 Řešení inverzní kinematiky

/\*\* 1 \* @brief Inverzní kinematika se řeší odkazem na zadanou polohu kloubu. 2 3 \* (aparam [in] typ 0- absolutní pozice (základní M ame), 1- inkrementální pozice (základní M ame), 2-*·→inkrementální pozice (nástroj M ame)* 4 \* @param [in] desc pos Kartézská pozice 5 \* @param [in] joint pos reMeference poloha kloubu 6 \* @param [out] joint pos Poloha kloubu 7 \* @*return* Kód chyby 8 \*/ 9 errno\_t GetInverseKinRef(int type, DescPose \*desc\_pos, JointPos \*joint\_pos\_ref,\_  $\leftrightarrow$  JointPos \*joint\_pos);

# 2.1.7.11 Řešení inverzní kinematiky

/\*\* 1 \* @brief Při řešení inverzní kinematiky se odkazuje na zadanou polohu kloubu. 2 *→určit,zda existuje řešení* 3 \* (aparam [in] typ 0- absolutní pozice (základní M ame), 1- inkrementální pozice (základní M ame), aparam [in], desc pos Kartézská pozice 4 \* aparam [in] joint pos reMeference poloha kloubu 5 @param [out] výsledek 0- žádné řešení, 1-řešení 6 (pokračování na další straně)

<sup>7</sup> \* @return Kód chyby

# 2.1.7.12 Řešení dopředné kinematiky

1	/**
2	* @brief Řešení dopředné kinematiky
3	* @param [in] joint_pos Poloha kloubu
4	* @param [out] desc_pos Kartézská pozice
5	* @return Kód chyby
6	*/
7	errno_tGetForwardKin(JointPos *joint_pos, DescPose *desc_pos);

# 2.1.7.13 Zjištění aktuálního kloubního momentu

1 /\*\*
2 % @brief Získání aktuálního kloubního momentu
3 % @param [in] příznak 0- blocking, 1- non-blocking
4 % @param [out] krouticí momenty Kloubový krouticí moment
5 % @return Kód chyby
7 \*/
errno\_t GetJointTorques(uint8\_t flag, float torques[6]);

#### 2.1.7.14 Zjištění hmotnosti aktuálního nákladu

/\*\*
/\*\*
\* @brief Zjistĺ hmotnost aktuálního nákladu
\* @param [in] příznak 0- blocking, 1- non-blocking
\* @param [out] weight Hmotnost nákladu, jednotka: kg
\* @return Kód chyby
\*/
errno\_t GetTargetPayload(uint8\_t flag, float \*weight);

#### 2.1.7.15 Zjištění středu hmotnosti aktuálního nákladu

1 /\*\*
2 /\*\*
2 @brief Zjištění středu hmotnosti aktuálního zatížení
3 \* @param [in] příznak 0- blocking, 1- non-blocking
4 \* @param [out] ozubené kolo Zatížení středu hmotnosti, jednotka: mm
5 \* @return Kód chyby
7 \*/
errno\_t GetTargetPayloadCog(uint8\_t flag, DescTran \*cog);

7

1

2

6

7

#### 2.1.7.16 Získání aktuálního souřadnicového systému nástroje

/\*\* 1 \* @brief Získání aktuálního souřadnicového systému nástroje 2 \* @param [in] příznak 0- blocking, 1- non-blocking 3 4 \* @param [out] desc pos Poloha souřadnic nástroje 5 \* @*return* Kód chyby 6 \*/ errno\_t GetTCPOffset(uint8\_t flag, DescPose \*desc\_pos);



/\*\* \* @brief Získejte aktuální pracovní dobu M 3 \* @param [in] příznak 0- blocking, 1- non-blocking 4 \* @param [out] desc pos Poloha souřadnicového systému obrobku 5 \* @*return* Kód chyby \*/

errno\_t GetWObjOffset(uint8\_t flag, DescPose \*desc\_pos);

#### 2.1.7.18 Získat měkký mezní úhel kloubu

/\*\* 1 \* @brief Získat měkký mezní úhel kloubu 2 3 \* @param [in] příznak 0- blocking, 1- non-blocking 4 \* @param [out] negativní Záporný mezní úhel, jednotka: deg 5 \* @param [out] pozitivní Kladný mezní úhel, jednotka: deg 6 \* @return Kód chyby 7 \*/ 8

errno\_t GetJointSoftLimitDeg(uint8\_t flag, float negative[6], float positive[6]);

# 2.1.7.19 Získat systémový čas

/\*\* 1 \* @brief Získat systémový čas 2 \* @param [out] t ms jednotka: ms 3 4 \* @*return* Kód chyby \*/ 5 6 errno\_t\_GetSystemClock(float \*t\_ms);

#### 2.1.7.20 Získání aktuální konfigurace kloubů robota

```
1 /**
2 /**
2 @brief Získání aktuální konfigurace kloubů robota
3 @param [out] config Konfigurace společného prostoru, rozsah [0~7]
4 @return Kód chyby
5 //
6 errno_t GetRobotCurJointsConfig(int *config);
```

#### 2.1.7.21 Získání aktuální rychlosti

```
/**
* @brief Zjištění aktuální rychlosti robota
* @param [out] vel Jednotkou je mm/s
* @return Kód chyby
*/
errno t GetDefaultTransVel(float *vel);
```

#### 2.1.7.22 Dotaz, zda je pohyb robota dokončen

```
1 /**
2 * @brief Dotaz, zda je pohyb robota dokončen
3 * @param [out] state 0- neúplný, 1- dokončený
4 * @return Kód chyby
5 */
6 errno_t GetRobotMotionDone(uint8_t *state);
```

#### 2.1.7.23 Příklad kódu

```
#include <cstdlib>
1
   #include <iostream>
2
   #include <stdio.h>
3
   #include <cstring>
4
   #include <unistd.h>
5
   #include "MRobot.h"
6
   #include "RobotTypes.h"
7
8
   using namespace std;
9
10
   int main(void)
11
   {
12
                                          //Instituce objektu robota
       MRobot robot;
13
       robot.RPC("192.168.58.2");
                                          /Navázat komunikační spojení s robotem
14
    •→kontrolér
15
        float yangle, zangle;
16
        int flag = 0;
17
       JointPos j_deg, j_rad;
18
                                                                                    (pokračování na další straně)
```

```
(pokračování na předchozí straně)
```

```
DescPose tcp, flange, tcp_offset, wobj_offset;
19
        DescTran cog:
20
        int id;
21
        float torques[6] = \{0.0\};
22
        hmotnost plováku;
23
        float neg_deg[6]={0.0},pos_deg[6]={0.0};
24
        float t_ms;
25
        int config;
26
        float vel;
27
28
        memset(&j_deg, 0, sizeof(JointPos));
29
        memset(&j_rad, 0, sizeof(JointPos));
30
        memset(&tcp, 0, sizeof(DescPose));
31
        memset(&flange, 0, sizeof(DescPose));
32
        memset(&tcp_offset, 0, sizeof(DescPose));
33
        memset(&wobj_offset, 0, sizeof(DescPose));
34
        memset(&cog, 0, sizeof(DescTran));
35
36
        robot.GetRobotInstallAngle(&vangle, &zangle);
37
        printf("yangle:%f,zangle:%f\n", yangle, zangle);
38
39
        robot.GetActualJointPosDegree(flag, &j_deg);
40
        printf("joint pos deg:%f,%f,%f,%f,%f,%f,%f,%f,%f,n", j_deg,jPos[0],j_deg,jPos[1],j_deg.
41
    \rightarrow pos[2], j_deg. jPos[3], j_deg. jPos[4], j_deg. jPos[5]);
42
        robot.GetActualJointPosRadian(flag, &j_rad);
43
        printf("joint pos rad:%f,%f,%f,%f,%f,%f,%f,n", j_rad.jPos[0],j_rad.jPos[1],j_rad.
44
    \rightarrow pos[2], j_rad. jPos[3], j_rad. jPos[4], j_rad. jPos[5]);
45
        robot.GetActualTCPPose(flag, &tcp);
46
        printf("tcp pose:%f,%f,%f,%f,%f,%f,%f\n", tcp.tran.x, tcp.tran.y, tcp.tran.z, tcp.rpy.
47
    →rx, tcp.rpy.ry, tcp.rpy.rz);
48
        robot.GetActualToolFlangePose(flag, &flange);
49
        printf("flange pose:%f,%f,%f,%f,%f,%f,%f,n", flange.tran.x, flange.tran.y, flange.tran.
50
    →z, flange.rpy.rx, flange.rpy.ry, flange.rpy.rz);
51
        robot.GetActualTCPNum(flag, &id);
52
        printf("tcp num:%d\n", id);
53
54
        robot.GetActualWObjNum(flag, &id);
55
        printf("wobj num:%d\n", id):
56
57
        robot.GetJointTorques(flag, torques);
58
        printf("točivé momenty:%f,%f,%f,%f,%f,%f,%f\n", torques[0],torques[1],torques[2],torques[3],
59
    →kroutici momenty [4], kroutící momenty [5]);
60
        robot.GetTargetPayload(flag, &weight);
61
        printf("hmotnost nákladu:%f\n", hmotnost);
62
63
        robot.GetTargetPayloadCog(flag, &cog);
64
        printf("payload cog:%f,%f,%f\n",cog.x, cog.y, cog.z);
65
                                                                                     (pokračování na další straně)
```

66	
67	robot.GetTCPOffset(flag, &tcp_offset);
68	printf("tcp offset:%f,%f,%f,%f,%f,%f, <b>%f\n</b> ", tcp_offset.tran.x,tcp_offset.tran.y,tcp_
	→offset.tran.z,tcp_offset.rpy.rx,tcp_offset.rpy.ry,tcp_offset.rpy.rz);
69	what Cattwol: Offert/flag, 0 and 1; a ffert)
70	robot.GetWObjOIIset(Ilag, &Wobj_OIIset);
71	printf("wobj offset:%i,%i,%i,%i,%i,%f,%t\n", wobj_offset.tran.x,wobj_offset.tran.y,wobj_
70	→offset.tran.z,wobj_offset.rpy.rx,wobj_offset.rpy.ry,wobj_offset.rpy.rz);
72	robot GetIointSoftLimitDeg(flag_neg_deg_pos_deg).
74	printf("neg limit deg:%f.%f.%f.%f.%f.%f.%f.%f.h".neg deg[0].neg deg[1].neg deg[2].neg
	(→deg[3],neg_deg[4],neg_deg[5]);
75	printf("pos limit deg:%f,%f,%f,%f,%f,%f,%f, <b>%f\n</b> ",pos_deg[0],pos_deg[1],pos_deg[2],pos_
	$\leftrightarrow$ deg[3],pos_deg[4],pos_deg[5]);
76	
77	robot.GetSystemClock(&t_ms);
78	printf("systémové hodiny: <b>%f\n</b> ", t_ms);
79	robot GetBabot(urIginteConfig(&config);
80	
81 82	printf("joint config:%d\n", config);
83	robot.GetDefaultTransVel(&vel);
84	printf("trans vel:%f\n", vel):
85	
86	návrat 0;
87	3

# 2.1.8 Opakování trajektorie

#### 2.1.8.1 Nastavení parametrů nahrávání stopy

```
/**
1
   * @brief Nastavení parametrů nahrávání stopy
2
3
   * @param [in] type Datový typ záznamu, 1- pozice kloubu
4
   * @param [in] name Název souboru stopy
5
   * aparam [in] period ms Perioda vzorkování dat, pevná hodnota 2ms nebo 4ms nebo 8ms
6
   * @param [in] di choose DI Select, bit0 až bit7 odpovídá ovládacímu poli DI0 až DI7,
   ↔ bit8 až bit9 odpovídá konci DI0 až DI1, 0- nevybírat, 1- vybrat
7
   * @param [in]
                             do chooseDO select, bit0~bit7 odpovídá ovládacímu poli DO0~DO7,
    bit8~
8
   ↔ bit9 odpovídá konci DO0~DO1, 0- nevybírat, 1- vybrat
q
   * @return Kód chyby
10
   */
   errno_t SetTPDParam(int type, char name[30], int period_ms, uint16_t di_choose, uint16_
```

#### 2.1.8.2 Spuštění nahrávání stopy

1 2

3

4

5

6

7

8

9

/\*\*

- \* @brief Spuštění nahrávání skladby
- \* @param [in] type Datový typ záznamu, 1- pozice kloubu
- \* @param [in] name Název souboru stopy
- \* @param [in] period\_ms Perioda vzorkování dat, pevná hodnota 2ms nebo 4ms nebo 8ms
- \* @param [in] di\_choose DI Select, bit0 až bit7 odpovídá ovládacímu poli DI0 až DI7,
- ↔bit8 až bit9 odpovídá konci DI0 až DI1, 0- nevybírat, 1- vybrat
- \* @param [in] do\_chooseDO select,bit0~bit7 odpovídá ovládacímu poli DO0~DO7, bit8~
- ↔bit9 odpovídá konci D00~D01, 0- nevybírat, 1- vybrat
- 10 \* @*return Kód chyby*

errno\_t <u>SetTPDStart(int</u> type, char name[30], int period\_ms, uint16\_t di\_choose, uint16\_

# 2.1.8.3 Zastavení nahrávání stopy

1 /\*\*
2 \* @brief Zastavení nahrávání stopy
3 \* @return Kód chyby
4 \*/
5 errno\_t\_\_SetWebTPDStop();

## 2.1.8.4 Vymazat záznam

1 /\*\*
2 \* @brief Vymazat záznam o činnosti
3 \* @param [in] name Název souboru stopy
4 \* @return Kód chyby
5 \*/

errno\_t \_\_SetTPDDelete(<mark>char</mark>\_name[30]);

#### 2.1.8.5 Příklad kódu

```
#include <cstdlib>
1
   #include <iostream>
2
   #include <stdio.h>
3
   #include <cstring>
4
   #include <unistd.h>
5
   #include "MRobot.h"
6
   #include "RobotTypes.h"
7
8
   using namespace std;
9
10
   int main(void)
11
   {
12
        Robot MRobot;
                                            //Instituce objektu robota
13
                                                                                        (pokračování na další straně)
```

```
/Navázat komunikační spojení s robotem
       robot.RPC("192.168.58.2");
14
      →kontrolér
15
       int type = 1;
16
        char name[30] = "tpd2023";
17
        int period_ms = 4; uint16 t
18
        di_choose = 0; uint16 t
19
       do_choose = 0;
20
21
       robot.SetTPDParam(type, name, period_ms, di_choose, do_choose);
22
23
       robot.Mode(1);
24
        sleep(1);
25
       robot.DragTeachSwitch(1);
26
       robot.SetTPDStart(type, name, period_ms, di_choose, do_choose);
27
       sleep(30);
28
       robot.SetWebTPDStop();
29
       robot.DragTeachSwitch(0);
30
31
       //robot.SetTPDDelete(name);
32
33
       vrátit 0;
34
   }
35
```

#### 2.1.8.6 Přednačítání trajektorie

```
1 /**
2 * @brief Přednačítání trajektorie
3 * @param [in] name Název souboru stopy
4 * @return Kód chyby
5 */
6 errno_t LoadTPD(char name[30]);
```

#### 2.1.8.7 Opakování trajektorie

```
/**
1
   * @brief Opakování trajektorie
2
3
   * @param [in] name Název souboru stopy
4
   * @param [in] blend 0- není hladký, 1- hladký
5
   * @param [in] ovl Procento škálování rychlosti, rozsah [0~100]
6
   * @return Kód chyby
7
   */
8
   errno_t MoveTPD(char name[30], uint8_t blend, float ovl);
```

#### 2.1.8.8 Příklad kódu

```
#include <cstdlib>
1
   #include <iostream>
2
   #include <stdio.h>
3
   #include <cstring>
4
   #include <unistd.h>
5
   #include "MRobot.h"
6
   #include "RobotTypes.h"
7
8
   using namespace std;
9
10
   int main(void)
11
   Ł
12
                                          //Instituce objektu robota
        MRobot robot;
13
       robot.RPC("192.168.58.2");
                                          /Navázat komunikační spojení s robotem
14
    •→kontrolér
15
        char name[30] = "tpd2023";
16
        int tool = 1;
17
       int user = 0; float
18
       vel = 100.0; float
19
       acc = 100.0; float
20
        ovl = 100.0:
21
        float blendT = -1.0:
22
        int config = -1;
23
       uint8_t blend = 1;
24
25
       DescPose desc_pose;
26
       memset(&desc_pose, 0, sizeof(DescPose));
27
28
       desc_pose.tran.x = -378.9;
29
       desc_pose.tran.y = -340.3;
30
        desc_pose.tran.z = 107.2;
31
       desc_pose.rpy.rx = 179.4;
32
        desc_pose.rpy.ry = -1.3;
33
       desc_pose.rpy.rz = 125.0;
34
35
       robot.LoadTPD(name);
36
       robot.MoveCart(&desc_pose, tool, user, vel, acc, ovl, blendT, config);
37
       robot.MoveTPD(name, blend, ovl);
38
39
       vrátit 0;
40
   }
41
```

# 2.1.9 Použití programu WebAPP

#### 2.1.9.1 Nastavení automatického načítání výchozího programu úloh při spuštění.

/\*\* 1 \* @brief 🛛 Nastavení výchozího programu Úlohy, který se automaticky načte při spuštění. 2 \* @param [in] flag
 0- boot automatické
 ·→ automatické načtení výchozího programu 0- boot automaticky nenačte výchozí program, 1- boot, 3 \* @param [in] program name Název a cesta k programu úlohy, například /Muser/movej.lua, 👝 4 *↔kde /Muser/ je pevná cesta* \* @,return Kód chyby 5 \*/ 6 7 errno\_t LoadDefaultProgConfig(uint8\_t flag, char program\_name[64]);

#### 2.1.9.2 Načtení zadaného programu úlohy

1	/**
2	* @brief Načtení zadaného pracovního programu
3	* @param [in] program_name Název a cesta k programu úlohy, například /Muser/movej.lua,
	<i>·→kde /Muser/ je pevná cesta</i>
4	* @return Kód chyby
5	*/
6	errno_t ProgramLoad( <mark>char</mark> program_name[64]);

#### 2.1.9.3 Získat název načteného programu úlohy

1 /\*\*

a @brief Získat název načteného programu úlohy
3 @param [out] program\_name Název a cesta k programu úlohy, například /Muser/movej.lua, \_\_\_\_\_
->kde /Muser/ je pevná cesta

4 a @return Kód chyby

\*/
6 errno\_t GetLoadedProgram(char program\_name[64]);

#### 2.1.9.4 Získání čísla řádku aktuálního programu úlohy robota

```
1 /**
2 @brief Získání čísla řádku aktuálního programu úlohy robota
3 @param [out] řádek číslo řádku
4 @return Kód chyby
5 %/
6 errno_t GetCurrentLine(int *line);
```

## 2.1.9.5 Spustit aktuálně načtený program úlohy

```
/**
* @brieMun aktuálně načtený pracovní program
* @return Kód chyby
*/
errno t ProgramRun();
```

# 2.1.9.6 Pozastavení aktuálně spuštěného programu úlohy

```
1 /**
2 * @brief Pozastavení aktuálně běžícího programu úlohy
3 * @return Kód chyby
4 */
5 errno_t ProgramPause();
```

# 2.1.9.7 Obnovení aktuálně pozastaveného pracovního programu

```
1 /**
2 /**
3 @brieMesume aktuálně pozastavený program pracovních míst
3 @return Kód chyby
4 */
5 errno t ProgramResume();
```

# 2.1.9.8 Ukončí aktuálně spuštěný program úlohy

```
1 /**
2 % @brief Ukončí aktuálně spuštěný program úlohy
3 % @return Kód chyby
4 */
5 errno_t ProgramStop();
```

# 2.1.9.9 Získání stavu provádění programu úlohy robota

```
1 /**

2 * @brief Získání stavu provádění programu úlohy robota
3 * @param [out] state 1- zastavení programu nebo žádný program neběží, 2- program běží, 3- ________
→ pauza programu
4 * @return Kód chyby
*/
6 errno_t GetProgramState(uint8_t *state);
```

#### 2.1.9.10 Příklad kódu

```
#include <cstdlib>
1
   #include <iostream>
2
   #include <stdio.h>
3
   #include <cstring>
4
   #include <unistd.h>
5
   #include "MRobot.h"
6
   #include "RobotTypes.h"
7
8
   using namespace std;
9
10
   int main(void)
11
   Ł
12
                                          //Instituce objektu robota
        MRobot robot;
13
       robot.RPC("192.168.58.2");
                                          /Navázat komunikační spojení s robotem
14
    •→kontrolér
15
        char program_name[64] = "/Muser/ptps.lua";
16
        char loaded_name[64] = "";
17
       uint8_t state;
18
        int line;
19
20
       robot.Mode(0);
21
        robot.ProgramLoad(název_programu);
22
       robot.ProgramRun();
23
       sleep(5);
24
       robot.ProgramPause();
25
        robot.GetProgramState(&state);
26
        printf("stav programu:%u\n", state);
27
       robot.GetCurrentLine(&line);
28
        printf("aktuální řádek:%d\n", line);
29
       robot.GetLoadedProgram(loaded_name);
30
        printf("název programu:%s\n", loaded_name);
31
       sleep(5);
32
       robot.ProgramResume();
33
        sleep(5);
        robot.ProgramStop();
34
       sleep(2);
35
36
       vrátit 0;
37
38
   }
39
```

# 2.1.10 Periferní

#### 2.1.10.1 Konfigurace chapadla

/\*\* 1 \* @brief Konfigurace chapadla 2 \* @param [ve] společnostiVýrobce drápů, bude určeno 3 \* @param [in] zařízení Číslo zařízení, zatím nepoužité. Výchozí hodnota je 0 4 \* @param [in] softvesion Verze softwaru. Hodnota se nepoužívá. Výchozí hodnota je... 5  $\leftrightarrow 0$ \* @param [in] bus Zařízení je připojeno k terminálové sběrnici a není p o u ž í v á n o. The\_ 6  $\leftrightarrow _{vychozi} hodnota je 0$ \* @*return* Kód chyby 7 \*/ 8 errno\_t SetGripperConfig(int company, int device, int softvesion, int bus); 9

#### 2.1.10.2 Získání konfigurace chapadla

1	/**
2	* @brief Získání konfigurace chapadla
3	* @param [ve] společnostiVýrobce drápů, bude určeno
4	* @param [in] zařízení Číslo zařízení, zatím nepoužité. Výchozí hodnota je 0
5	* @param [in] softvesion Verze softwaru. Hodnota se nepoužívá. Výchozí hodnota je
	$\leftrightarrow 0$
6	* @param [in] bus Zařízení je připojeno k terminálové sběrnici a není p o u ž í v á n o . The
	$\leftrightarrow$ výchozí hodnota je 0
7	* @return Kód chyby
8	*/
9	errno_t GetGripperConfig(int *company, int *device, int *softvesion, int *bus);

#### 2.1.10.3 Aktivace chapadla

#### 2.1.10.4 Řídicí chapadlo

1

/\*\*

\* @brief Řídicí chapadlo
\* @param [in] index číslo chapadla
\* @param [in] pos Procento pozice, rozsah[0~100]
\* @param [in] vel Procento rychlosti, rozsah[0~100]
\* @param [in] force Procento točivého momentu, rozsah[0~100]
\* @param [in] max\_time Maximální doba čekání, rozsah[0~30000], jednotka: ms
\* @param [in] blok 0- blocking, 1- non-blocking
\* @return Kód chyby
\*/
errno\_t MoveGripper(int index, int pos, int vel, int force, int max\_time, uint8\_t\_

#### 2.1.10.5 Získání stavu pohybu chapadla

```
1 /**
2 * @brief Získání stavu pohybu chapadla
3 * @param [out] fault 0- bez chyby, 1- chyba
4 * @param [out] staus 0- pohyb nedokončen, 1- pohyb dokončen
5 * @return Kód chyby
6 */
7 errno_t GetGripperMotionDone(uint8_t *fault, uint8_t *status);
```

#### 2.1.10.6 Příklad kódu

	7		
1	<i>#include <cstdlib></cstdlib></i>		
2	<i>#include <iostream></iostream></i>		
3	<i>#include <stdio.h></stdio.h></i>		
4	<i>#include <cstring></cstring></i>		
5	<i>#include <unistd.h></unistd.h></i>		
6	#include "MRobot.h"		
7	#include "RobotTypes.h"		
8			
9	using namespace <pre>std;</pre>		
10			
11	int main(void)		
12	{		
13	MRobot robot;	//Instituce objektu robota	
14	robot.RPC("192.168.58.2");	/Navázat komunikační spojení s robotem,	
	<i>•→kontrolér</i>		
15			
16	int company = 4;		
17	int device = 0;		
18	int softversion = 0;		
19	int bus = 1;		
20	int index = 1;		
21	$\frac{1111}{1111} \text{ act} = 0;$		
22	$\operatorname{int} \operatorname{max\_time} = 30000;$		
			pokračování na další straně)

23	<pre>uint8_t block = 0;</pre>
24	<pre>uint8_t status, fault;</pre>
25	and at SatOnian and and in (a summary density of the unity has)
26	robot.SetGripperConfig(company, device, softversion, bus);
27	robot GetGrinnerConfig(&společnost &zařízení &softverze &shěrnice)
28	printf("konfigurace gripperu:%d.%d.%d.%d.%d.n", společnost, zařízení, softverze,
29	sběrnice);
30	
32	robot.ActGripper(index, act);
33	sleep(1);
34	act = 1; robot.ActGripper(index,
35	act), steep(z),
36	robot.MoveGripper(index, 100, 50, 50, max_time, block);
37	sleep(3);
38	robot.MoveGripper(index, 0, 50, 0, max_time, block);
39	
40	robot.GetGripperMotionDone(&fault, &status); printf("stav
41	pohybu:%u,%u\n", fault, status);
42	vrátit O:
43	viacit 0,
45	3
-	

# 2.1.11 Kontrola síly

# 2.1.11.1 Konfigurace snímače síly

1	/**	
2	* @brief	Nakonfigurovaný snímač síly
3	* @param	[ve] společnosti Výrobce snímačů síly, 17-Kunwei
4	* @param	Technology [in] device Číslo zařízení, dosud nepoužité.
5	* @param	Výchozí hodnota je 0
6	<→0 * @param	[in] softvesion Verze softwaru. Hodnota se nepoužívá. Výchozí hodnota je [in] bus Zařízení je připojeno k terminálové sběrnicí a není p o ú ž i v á n o . The
	ightarrow výchozí h	nodnota je 0
7	* @return	Kód chyby
8	*/	
9	errno_t	FT_SetConfig(int company, int device, int softvesion, int bus);

# 2.1.11.2 Získání konfigurace snímače síly

1	/**	
2	* @brief	Získání konfigurace snímače síly
3	* @param	[ve] společnosti Výrobce snímače síly, bude určeno
4	* @param	[in] zařízení Číslo zařízení, zatím nepoužité. Výchozí hodnota je 0
5	* @param	[in] softvesion Verze softwaru. Hodnota se nepoužívá. Výchozí hodnota je
	$\longleftrightarrow 0$	
6	* @param	[in] bus Zařízení je připojeno k terminálové sběrnici a není p o u ž í v á n o . The
		(pokračování na další straně)

```
' · · · vychozi hodnota je 0
' * @return Kód chyby
*/
errno_t FT_GetConfig(int *company, int *device, int *softvesion, int *bus);
```

#### 2.1.11.3 Aktivace snímače síly

1 /\*\*
2 \* @brief Aktivace snímače síly
3 \* @param [in] act 0- reset, 1- activate
4 \* @return Kód chyby
5 \*/
6 errno t FT Activate(uint8\_t act);

#### 2.1.11.4 Kalibrace snímače síly

1 /\*\*
2 % @brief Kalibrace snímače síly
3 \* @param [in] act 0- odstranění nuly, 1- oprava nuly
4 \* @return Kód chyby
5 6
6 errno\_t FT\_SetZero(uint8\_t act);

#### 2.1.11.5 Příklad kódu

```
#include <cstdlib>
1
   #include <iostream>
2
   #include <stdio.h>
3
   #include <cstring>
4
   #include <unistd.h>
5
6
   #include "MRobot.h"
7
   #include "RobotTypes.h"
8
9
   using namespace std;
10
11
   int main(void)
12
   Ł
13
                                           //Instituce objektu robota
        MRobot robot;
14
        robot.RPC("192.168.58.2");
                                           /Navázat komunikační spojení s robotem
15
    ⊶kontrolér
16
        int company = 17;
17
        int device = 0;
18
        int softversion = 0;
19
        int bus = 1;
20
        int index = 1;
21
                                                                                     (pokračování na další straně)
```

22	<b>int</b> act = 0;	
23	robot FT SotConfig(company dovice softwarsion bus);	
24	sleen(1).	
25	robot FT GetConfig(&company &device &softwarsion &bus):	
20	printf("FT config.%d %d %d %d\n" company device softversion bus).	
21	sleen(1).	
20	sicep(1),	
30	robot.FT_Activate(act):	
31	sleep(1);	
32	act = 1;	
33	robot.FT_Activate(act);	
34	sleep(1);	
35		
36	robot.SetLoadWeight(0.0);	
37	sleep(1);	
38	DescTran coord;	
39	memset(&coord, 0, sizeof(DescTran));	
40	robot.SetLoadCoord(&coord);	
41	sleep(1);	
42	robot.FT_SetZero(0);	
43	sleep(1);	
44	ForceTorque ft.	
45	memset(&ft 0 sizeof(ForceTorque)):	
46	robot FT GetForceTorqueOrigin(&ft)	
47	$\operatorname{printf}("ft \operatorname{origin}) \circ f \circ $	
40	robot FT_SetZero(1):	
50	sleep(1):	
51	memset(&ft, 0, sizeof(ForceTorque));	
52	printf("ft rcs:%f,%f,%f,%f,%f,%f,%f\n",ft.fx,ft.fy,ft.fz,ft.tx,ft.ty,ft.tz);	
53		
54	vrátit 0;	
55	\ }	

# 2.1.11.6 Nastavení referenčního souřadného systému snímače síly

```
1 /**
2 * @brief Nastavení referenčního souřadného systému snímače síly
3 * @param [in] ref 0- tool M ame, 1- base M ame
4 * @return Kód chyby
5 */
6 errno_t FT_SetRCS(uint8_t ref);
```

### 2.1.11.7 Záznam o identifikaci hmotnosti nákladu

```
1 /**
2 /**
2 @brief Záznam o identifikaci hmotnosti nákladu
3 @param [in] id Číslo souřadného systému senzoru, rozsah [1~14]
4 @return Kód chyby
5 /
6 errno_t FT_PdIdenRecord(int id);
```

# 2.1.11.8 Výpočet identifikace hmotnosti nákladu

```
1 /**
2 * @brief Výpočet identifikace hmotnosti nákladu
3 * @param [out] weight Hmotnost nákladu, jednotka: kg
4 * @return Kód chyby
5 */
6 errno_t FT_PdIdenCompute(float *weight);
```

# 2.1.11.9 Načtení identifikačního záznamu centroidu

```
1 /**
2 * @brief Načtení identifikačního záznamu centroidu
3 * @param [in] id Číslo souřadného systému senzoru, rozsah [1~14]
4 * @param [in] index Číslo bodu, rozsah [1~3]
5 * @return Kód chyby
6 */
7
```

<code>errno\_t FT\_PdCogIdenRecord(int id, int index);</code>

#### 2.1.11.10 Výpočet identifikace centroidů zatížení

```
1 /**
2 * @brief Výpočet identifikace centroidů zatížení
3 * @param [out] cog Zátěžový střed, jednotka: mm
4 * @return Kód chyby
5
6 errno_t FT_PdCogIdenCompute(DescTran *cog);
```

#### 2.1.11.11 Příklad kódu

```
      1
      #include <cstdlib>

      2
      #include <iostream>

      3
      #include <stdio.h>

      4
      #include <cstring>

      5
      #include <unistd.h>

      6
      #include "MRobot.h"

      7
      #include "RobotTypes.h"
```

(pokračování na další straně)

```
using namespace std;
9
10
   int main(void)
11
   Ł
12
        Robot MRobot;
                                          //Instituce objektu robota
13
        robot.RPC("192.168.58.2");
                                          /Navázání komunikačního spojení s robotem
14
      →kontrolér
15
       hmotnost plováku;
16
17
       DescPose tcoord, desc_p1, desc_p2, desc_p3;
18
       memset(&tcoord, 0, sizeof(DescPose));
19
        memset(&desc_p1, 0, sizeof(DescPose));
20
       memset(&desc_p2, 0, sizeof(DescPose));
21
        memset(&desc_p3, 0, sizeof(DescPose));
22
23
       robot.FT_SetRCS(0);
24
       sleep(1);
25
26
       tcoord.tran.z = 35.0;
27
        robot.SetToolCoord(10, &tcoord, 1, 0);
28
       sleep(1);
29
       robot.FT_PdIdenRecord(10);
30
        spánek(1);
31
       robot.FT_PdIdenCompute(&weight);
32
        printf("hmotnost nákladu:%f\n", hmotnost);
33
34
        desc_p1.tran.x = -160.619;
35
        desc_p1.tran.y = -586.138;
36
        desc_p1.tran.z = 384.988;
37
        desc_p1.rpy.rx = -170.166;
38
        desc_p1.rpy.ry = -44.782;
39
        desc_p1.rpy.rz = 169.295;
40
41
        desc_p2.tran.x = -87.615;
42
        desc_p2.tran.y = -606.209;
43
        desc_p2.tran.z = 556.119;
44
        desc_p2.rpy.rx = -102.495;
45
        desc_p2.rpy.ry = 10.118;
46
        desc_p2.rpy.rz = 178.985;
47
48
        desc_p3.tran.x = 41.479;
49
        desc_p3.tran.y = -557.243;
50
        desc_p3.tran.z = 484.407;
51
        desc_p3.rpy.rx = -125.174;
52
        desc_p3.rpy.ry = 46.995;
53
        desc_p3.rpy.rz = -132.165;
54
55
       robot.MoveCart(&desc_p1, 9, 0, 100.0, 100.0, 100.0, -1.0, -1);
56
       spánek(1);
57
        robot.FT_PdCogIdenRecord(10, 1);
58
       robot.MoveCart(&desc_p2, 9, 0, 100.0, 100.0, 100.0, -1.0, -1);
59
                                                                                    (pokračování na další straně)
```
```
sleep(1);
60
       robot.FT_PdCogIdenRecord(10, 2);
61
       robot.MoveCart(&desc_p3, 9, 0, 100.0, 100.0, 100.0, -1.0, -1);
62
       sleep(1);
63
       robot.FT_PdCogIdenRecord(10, 3);
64
       sleep(1);
65
       DescTran cog;
66
       memset(&cog, 0, sizeof(DescTran));
67
       robot.FT_PdCogIdenCompute(&cog);
68
       printf("cog:%f,%f,%f\n",cog.x, cog.y, cog.z);
69
70
       vrátit 0;
71
   }
72
```

### 2.1.11.12 Získání údajů o síle/kroutícím momentu v referenčním souřadném systému

1	/**
2	* @brief Získání údajů o síle/točivém momentu v referenčním souřadném systému
3	* @param [out] ft Force/torquefx,fy,fz,tx,ty,tz
4	* @return Kód chyby
5	*/
6	errno t FT GetForceTorqueRCS(ForceTorque *ft);

## 2.1.11.13 Získání nezpracovaných údajů o síle/otáčivém momentu ze snímače síly

```
1 /**
2 % @brief Získání nezpracovaných údajů o síle/točivém momentu ze snímače síly
3 % @param [out] ft Force/torquefx,fy,fz,tx,ty,tz
4 * @return Kód chyby
5 */
6 errno_t_FT_GetForceTorqueOrigin(ForceTorque *ft);
```

## 2.1.11.14 Ochranný systém proti nárazu

/**
* @brief Kolizní ochrana
<ul> <li>* @param [in] flag 0- Zakázat ochranu proti kolizím. 1- Povolit ochranu proti kolizím</li> <li>* @param [in] sensor_id Číslo silového senzoru</li> </ul>
* @param [in] select Vybere šest stupňů volnosti, zda má být detekována kolize, 0 ↔žádná detekce, 1- detekce
* @param [in] ft Rázová síla/otáčivý momentfx, fy, fz, tx, ty, tz
* @param [in] max_threshold Maximální prahová hodnota
* @param [in] min_threshold Minimální prahová hodnota
* @note Rozsah detekce síly/motoru(ft-min_threshold, ft+max_threshold)
* @return Kód chyby
*/
errno_t FT_Guard(uint8_t flag, int sensor_id, uint8_t select[6], ForceTorque *ft, floa
max_threshold[6], float min_threshold[6]);

#### 2.1.11.15 Příklad kódu

```
#include <cstdlib>
 1
         #include <iostream>
 2
        #include <stdio.h>
 3
        #include <cstring>
 4
        #include <unistd.h>
 5
        #include "MRobot.h"
 6
        #include "RobotTypes.h"
 7
 8
        using namespace std;
 9
10
        int main(void)
11
        {
12
                   Robot MRobot;
                                                                                                        //Instituce objektu robota
13
                   robot.RPC("192.168.58.2");
                                                                                                        /Navázání komunikačního spojení s robotem
14
          ↔kontrolér
15
                   uint8_t flag = 1;
16
                   uint8_t sensor_id = 1;
17
                   uint8_t select[6] = {1,1,1,1,1,1,1;;
18
                   float max_threshold[6] = \{10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10
19
                   float min_threshold[6] = {5.0,5.0,5.0,5.0,5.0,5.0,5.0};
20
21
                   ForceTorque ft;
22
                   DescPose desc_p1, desc_p2, desc_p3;
23
                   memset(&ft, 0, sizeof(ForceTorque));
24
                   memset(&desc_p1, 0, sizeof(DescPose));
25
                   memset(&desc_p2, 0, sizeof(DescPose));
26
                   memset(&desc_p3, 0, sizeof(DescPose));
27
28
                   desc_p1.tran.x = -160.619;
29
                   desc_p1.tran.y = -586.138;
30
                   desc_p1.tran.z = 384.988;
31
                   desc_p1.rpy.rx = -170.166;
32
                    desc_p1.rpy.ry = -44.782;
33
                   desc_p1.rpy.rz = 169.295;
34
35
                   desc_p2.tran.x = -87.615;
36
                    desc_p2.tran.y = -606.209;
37
                   desc_p2.tran.z = 556.119;
38
                   desc_p2.rpy.rx = -102.495;
39
                   desc_p2.rpy.ry = 10.118;
40
                   desc_p2.rpy.rz = 178.985;
41
42
                   desc_p3.tran.x = 41.479;
43
                   desc_p3.tran.y = -557.243;
44
                   desc_p3.tran.z = 484.407;
45
                   desc_p3.rpy.rx = -125.174;
46
                   desc_p3.rpy.ry = 46.995;
47
                   desc_p3.rpy.rz = -132.165;
48
49
                   robot.FT_Guard(flag, sensor_id, select, &ft, max_threshold, min_threshold);
50
```

```
(pokračování na další straně)
```

```
si robot.MoveCart(&desc_p1,9,0,100.0,100.0,-1.0,-1);
si robot.MoveCart(&desc_p2,9,0,100.0,100.0,-1.0,-1);
si robot.MoveCart(&desc_p3,9,0,100.0,100.0,-1.0,-1);
flag = 0;
si robot.FT_Guard(flag, sensor_id, select, &ft, max_threshold, min_threshold);
vrátit 0;
si }
```

### 2.1.11.16 Řízení konstantní síly

/\*\* 1 \* @brief Stálá kontrola síly 2 3 \* @param [in] příznak 0- vypnutí konstantní kontroly síly, 1- zapnutí konstantní kontroly síly \* @param [in] sensor id Číslo silového senzoru 4 5 \* @param [in] select – Vybere šest stupňů volnosti, zda má být detekována kolize, 0-\_ ·→žádná detekce, 1- detekce 6 \* @param [in] ft Rázová síla/otáčivý momentfx, fy, fz, tx, ty, tz 7 \* @param [in] ft pid Parametr Force pid, parametr Torque pid 8 \* @param [in] adj sign Adaptivní řízení start-stop, 0- vypnuto, 1- zapnuto 9 \* @param [in] ILC sign ILC start stop control, 0- stop, 1- training, 2- operation 10 \* @param [in] Maximální vzdálenost nastavení, jednotka: mm 11 \* @param [in] Maximální úhel nastavení, jednotka: deg 12 \* @return Kód chyby 13 \*/ 14 errno\_t FT\_Control(uint8\_t flag, int sensor\_id, uint8\_t select[6], ForceTorque \*ft,

float ft\_pid[6], uint8\_t adj\_sign, uint8\_t ILC\_sign, float max\_dis, float max\_ang);

## 2.1.11.17 Příklad kódu

```
#include <cstdlib>
1
   #include <iostream>
2
   #include <stdio.h>
3
   #include <cstring>
4
   #include <unistd.h>
5
   #include "MRobot.h"
6
   #include "RobotTypes.h"
7
8
   using namespace std;
9
10
   int main(void)
11
   Ł
12
                                          //Instituce objektu robota
        MRobot robot;
13
        robot.RPC("192.168.58.2");
                                          /Navázat komunikační spojení s robotem
14
     →kontrolér
15
       uint8_t flag = 1;
16
        uint8_t sensor_id = 1;
17
        uint8 t select[6] = \{0,0,1,0,0,0,0\};
18
```

```
float ft_pid[6] = \{0.0005, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0\};
19
        uint8 t adj_sign = 0;
20
        uint8_t ILC_sign = 0;
21
        float max_dis = 100.0;
22
        float max_ang = 0,0;
23
24
        ForceTorque ft;
25
        DescPose desc_p1, desc_p2, offset_pos;
26
        JointPos j1,j2;
27
        ExaxisPos epos;
28
        memset(&ft, 0, sizeof(ForceTorque));
29
        memset(&desc_p1, 0, sizeof(DescPose));
30
        memset(&desc_p2, 0, sizeof(DescPose));
31
        memset(&offset_pos, 0, sizeof(DescPose));
32
        memset(&epos, 0, sizeof(ExaxisPos));
33
        memset(&j1, 0, sizeof(JointPos));
34
        memset(&j2, 0, sizeof(JointPos));
35
36
        j1 = \{-68.987, -96.414, -111.45, -61.105, 92.884, 11.089\};
37
        j2 = \{-107.596, -109.154, -104.735, -56.176, 90.739, 11.091\};
38
39
        desc_p1.tran.x = 62.795;
40
        desc_p1.tran.y = -511.979;
41
        desc_p1.tran.z = 291.697;
42
        desc_p1.rpy.rx = -179.545;
43
        desc_p1.rpy.ry = 3.027;
44
        desc_p1.rpy.rz = -170.039;
45
46
        desc_p2.tran.x = -294.768;
47
        desc_p2.tran.y = -503.708;
48
        desc_p2.tran.z = 233.158;
49
        desc_p2.rpy.rx = 179.799;
50
        desc_p2.rpy.ry = 0,713;
51
        desc_p2.rpy.rz = 151.309;
52
53
        ft.fz = -10,0;
54
55
        robot.MoveJ(&j1,&desc_p1,9,0,100.0,180.0,100.0,&epos,-1.0,0,&offset_pos);
56
        robot.FT_Control(flag, sensor_id, select, &ft, ft_pid, adj_sign, ILC_sign, max_dis,
57
    ,
→max_ang);
        robot.MoveL(&j2,&desc_p2,9,0,100.0,180.0,20.0,-1.0,&epos,0,0,&offset_pos);
58
        flag = 0;
59
        robot.FT_Control(flag, sensor_id, select, &ft, ft_pid, adj_sign, ILC_sign, max_dis,_
60
    , →max_ang;
61
        návrat 0;
62
   }
63
```

### 2.1.11.18 Spirálový průzkum

/\*\* 1

2

3

5

6

8

* (a)brief Spiralovy pruzka	ит	průzk	ový	Spirá	brief	(a)t	*
-----------------------------	----	-------	-----	-------	-------	------	---

- \* @param [in] rcs Reference M ame, 0- tool M ame, 1- base M ame
- 4 \* @param [in] dr Podávání na poloměr kruhu
- \* @param [in] ft Práh síly/kroutícího momentufx,fy,fz,tx,ty,tzrozsah[0~100]
- \* @param [in] max t ms Maximální doba průzkumu, jednotka: ms
- \* @param [in] max vel Maximální lineární rychlost, jednotka: mm/s
- \* @return Kód chyby

10

errno t FT SpiralSearch(int rcs, float dr, float ft, float max t ms, float max vel);

### 2.1.11.19 Rotační vkládání

/\*\* 1

2

3

4 5

6

7

8

9

12

- \* @brieMotarv vložení
- \* @param [in] rcs Reference M ame, 0- tool M ame, 1- base M ame
- \* @param [in] angVelRot Úhlová rychlost oMotation, jednotka: deg/s
- \* @param [in] ft Prahová hodnota síly/motorufx,fy,fz,tx,ty,tzrozsah[0~100]
- \* @param [in] max angle Maximální úhel natočení, jednotka: deg
- \* @param [in] orn Směr síly/motoru, 1- podél osy z, 2- kolem osy z
- \* @param [in] max\_angAcc Maximální rotační zrychlení, v deg/s^2, zatím nepoužité, 👝  $\longleftrightarrow$  výchozí hodnota je~0
- \* @param [in] rotorn Směr otáčení, 1- po směru hodinových ručiček, 2- proti směru 10 hodinových ručiček 11
  - \* @return Kód chyby

errno\_t FT\_RotInsertion(int rcs, float angVelRot, float ft, float max\_angle, uint8\_t\_

→om, float max\_angAcc, uint8 t rotorn);

2.1.11.20 Lineární vkládání

/\*\*

1

2

3

4 5

6

7

8

```
* @brief Lineární vkládání
```

- \* @param [in] rcs Reference M ame, 0- tool M ame, 1- base M ame
- \* @param [in] ft Prahová hodnota síly/motorufx, fy, fz, tx, ty, tzrozsah[0~100]
- \* @param [in] lin v Lineární rychlost, jednotka: mm/s
- \* @param [in] lin a Lineární zrychlení, jednotka: mm/s^2, zatím nepoužité
- \* @param [in] max dis Maximální vzdálenost vložení, jednotka: mm
- \* @param [in] linorn Směr vložení, 0- záporný, 1- kladný 9
- \* @return Kód chyby 10 \*/

11

errno\_t FT\_LinInsertion(int rcs, float ft, float lin\_v, float lin\_a, float max\_dis,\_\_ · Juint8 t linorn);

### 2.1.11.21 Příklad kódu

```
#include <cstdlib>
1
   #include <iostream>
2
   #include <stdio.h>
3
   #include <cstring>
4
   #include <unistd.h>
5
   #include "MRobot.h"
6
   #include "RobotTypes.h"
7
8
   using namespace std;
9
10
   int main(void)
11
   {
12
       Robot MRobot;
                                        //Instituce objektu robota
13
       robot.RPC("192.168.58.2");
                                        /Navázání komunikačního spojení s robotem
14
   ∝→kontrolér
15
       /Parametr konstantní sílv
16
       uint8_t status = 1; //Constant force control open sign, 0- vypnuto, 1- zapnuto
17
       int sensor_num = 1; //Číslo senzoru síly
18
       19
       uint8_t adj_sign = 0; //Adaptivní stav start-stop, 0- vypnuto, 1- zapnuto
20
                               //ILC control start stop state, 0- stop, 1- training, 2- real
       uint8_t ILC_sign = 0;
21
      →operace
       float max dis = 100.0: //Maximální vzdálenost nastavení
22
       float max_ang = 5.0;
                              //Maximální úhel nastavení
23
24
       ForceTorque ft:
25
       memset(&ft, 0, sizeof(ForceTorque));
26
27
       //Helix prozkoumat parametry
28
       int rcs = 0; //Reference M ame, 0- tool M ame, 1- base M ame
29
       float dr = 0,7; //Posuv o poloměru na otáčku, jednotka: mm
30
       float fFinish = 1.0; //Prahová hodnota síly nebo točivého momentu (0 až 100), jednotka: N nebo Nm
31
       float t = 60000.0; //Maximální doba průzkumu, jednotka: ms
32
       float vmax = 3.0; //Maximální lineární rychlost, jednotka: mm/s
33
34
       //Lineární vkládací parametr
35
       float force_goal = 20.0; //Prahová hodnota síly nebo točivého momentu (0 až 100), jednotka: N nebo Nm
36
       float lin_v = 0.0; //Lineární rychlost, jednotka: mm/s
37
       float lin_a = 0.0; //Lineární zrychlení, jednotka: mm/s^2, zatím se nepoužívá
38
       float disMax = 100.0; //Maximální vzdálenost vložení, v mm
39
       uint8_t linorn = 1; //Vložte směr, 1- kladný, 2- záporný
40
41
       //Rotační parametr vložení
42
       float angVelRot = 2.0; /Úhlová rychlost oMotation, v %
43
       float forceInsertion = 1.0; //Prahová hodnota síly nebo točivého momentu (0 až 100), v N nebo Nm
44
       int angleMax= 45; //Maximální úhel natočení, jednotka: °
45
       uint8_t orn = 1; //Směr síly1-fz,2-mz
46
       float angAccmax = 0.0; //Maximální úhlové zrychlení oMotation, jednotka: °/s^2, ne_...
47
      →použití
       uint8_t rotorn = 1; //Směr otáčení, 1- po směru hodinových ručiček, 2- proti směru hodinových ručiček
48
                                                                               (pokračování na další straně)
```

```
49
        uint8_t select1[6] = \{0, 0, 1, 1, 1, 1, 0\}; /Šest stupňů volnosti [fx, fy, fz, mx, my,
50
     mz], 0- nefunguje, 1- funguje
        ft.fz = -10,0;
51
        robot.FT_Control(status,sensor_num,select1,&ft,gain,adj_sign,ILC_sign,max_dis,max_
52
    <sub>→ang</sub>);
        robot.FT_SpiralSearch(rcs,dr,fFinish,t,vmax);
53
        status = 0:
54
        robot.FT_Control(status,sensor_num,select1,&ft,gain,adj_sign,ILC_sign,max_dis,max_
55
    <sub>'→ang</sub>);
56
        uint8 t select2[6] = \{1,1,1,0,0,0\};
                                                   //Šest stupňů volnosti [fx,fy,fz,mx,
57
    →my,mz], 0- nefunguje, 1- funguje
        gain[0] = 0.00005;
58
        ft.fz = -30,0;
59
        status = 1;
60
        robot.FT_Control(status,sensor_num,select2,&ft,gain,adj_sign,ILC_sign,max_dis,max_
61
      ng);
        robot.FT_LinInsertion(rcs.force_goal,lin_v,lin_a,disMax,linorn);
62
        status = 0;
63
        robot.FT_Control(status,sensor_num,select2,&ft,gain,adj_sign,ILC_sign,max_dis,max_
64
    →ang);
65
        uint8 t select3[6] = \{0.0.1.1.1.1.0\}:
                                                   //Šest stupňů volnosti [fx,fy,fz,mx,
66
    →my,mz], 0- nefunguje, 1- funguje
        ft.fz = -10,0;
67
        gain[0] = 0,0001;
68
        status = 1;
69
        robot.FT_Control(status,sensor_num,select3,&ft,gain,adj_sign,ILC_sign,max_dis,max_
70
    <sub>'→ang</sub>);
        robot.FT_RotInsertion(rcs,angVelRot,forceInsertion,angleMax,orn,angAccmax,rotorn);
71
        status = 0;
72
        robot.FT_Control(status,sensor_num,select3,&ft,gain,adj_sign,ILC_sign,max_dis,max_
73
    <sub>→ang</sub>);
74
        uint8_t select4[6] = {1,1,1,0,0,0};
                                                   //Šest stupňů volnosti [fx,fy,fz,mx,
75
    →my,mz], 0- nefunguje, 1- funguje
        ft.fz = -30,0;
76
        status = 1;
77
        robot.FT_Control(status,sensor_num,select4,&ft,gain,adj_sign,ILC_sign,max_dis,max_
78
      ang);
        robot.FT_LinInsertion(rcs,force_goal,lin_v,lin_a,disMax,linorn);
79
        status = 0;
80
        robot.FT_Control(status,sensor_num,select4,&ft,gain,adj_sign,ILC_sign,max_dis,max_
81
    <sub>(→ang</sub>);
82
        vrátit 0:
83
84
```

#### 2.1.11.22 Polohování na povrchu

1 2

4

7

8

9

10

11

12

/\*\*

- \* @brief Polohování na povrchu
- <sup>3</sup> \* @param [in] rcs Reference M ame, 0- tool M ame, 1- base M ame
- \* @param [in] dir Směr jízdy, 1- kladný, 2- záporný
- <sup>5</sup> \* @param [in] axis Osa pohybu, 1-x axis, 2-y axis, 3-z axis
  - \* @param [in] lin v Prozkoumejte lineární rychlost v mm/s
- \* @param [in] lin\_a Prozkoumejte lineární zrychlení, v mm/s^2, zatím se nepoužívá, výchozí hodnota je 0
- \* @param [in] max\_dis Maximální vzdálenost průzkumu, v mm
- \* @param [in] ft Akční práh ukončení síly/kroutícího momentufx, fy, fz, tx, ty, tz
- \* @return Kód chyby

errno\_t FT\_FindSurface(int rcs, uint8\_t dir, uint8\_t axis, float lin\_v, float lin\_a,

## float max\_dis, float ft);

### 2.1.11.23 Výpočeť polohy sťředové roviny začíná

/\*\*
 \* @brief Výpočet polohy středové roviny začíná
 \* @return Kód chyby
 \*/
 \*/
 \*/
 \*/

#### 2.1.11.24 Výpočet konců polohy ve střední rovině

/\*\*
 /\*\*
 @brief Výpočet konců polohy ve střední rovině
 @param [out] pos Mezipoloha roviny
 \* @return Kód chyby
 \*/
errno t FT CalCenterEnd(DescPose \*pos);

#### 2.1.11.25 Příklad kódu

```
#include <cstdlib>
1
2
   #include <iostream>
   #include <stdio.h>
3
   #include <cstring>
4
   #include <unistd.h>
5
   #include "MRobot.h"
6
   #include "RobotTypes.h"
7
8
   using namespace std;
9
10
   int main(void)
11
12
   Ł
        Robot MRobot;
                                            //Instituce objektu robota
13
                                                                                        (pokračování na další straně)
```

```
robot.RPC("192.168.58.2");
                                          /Navázání komunikačního spojení s robotem
14
      →kontrolér
15
       int rcs = 0;
16
        uint8_t dir = 1;
17
        uint8 t axis = 1;
18
        float lin_v = 3.0;
19
        float lin_a = 0,0;
20
        float maxdis = 50.0;
21
        float ft_goal = 2.0;
22
23
        DescPose desc_pos, xcenter, ycenter;
24
       ForceTorque ft;
25
        memset(&desc_pos, 0, sizeof(DescPose));
26
        memset(&xcenter, 0, sizeof(DescPose));
27
        memset(&ycenter, 0, sizeof(DescPose));
28
        memset(&ft, 0, sizeof(ForceTorque));
29
30
        desc_pos.tran.x = -230.959;
31
        desc_pos.tran.y = -364.017;
32
        desc_pos.tran.z = 217.5;
33
       desc_pos.rpy.rx = -179.004;
34
       desc_pos.rpy.ry = 0.002;
35
        desc_pos.rpy.rz = 89.999;
36
37
       ft.fx = -2,0;
38
39
       robot.MoveCart(&desc_pos, 9,0,100.0,100.0,100.0,-1.0,-1);
40
41
       robot.FT_CalCenterStart();
42
       robot.FT_FindSurface(rcs, dir, axis, lin_v, lin_a, maxdis, ft_goal);
43
       robot.MoveCart(&desc_pos, 9,0,100.0,100.0,100.0,-1.0,-1);
44
       robot.WaitMs(1000);
45
46
       dir = 2:
47
        robot.FT_FindSurface(rcs, dir, axis, lin_v, lin_a, maxdis, ft_goal);
48
       robot.FT_CalCenterEnd(&xcenter);
49
        printf("xcenter:%f,%f,%f,%f,%f,%f,%f\n",xcenter.tran.x,xcenter.tran.y,xcenter.tran.z,
50
    →xcenter.rpy.rx,xcenter.rpy.ry,xcenter.rpy.rz);
       robot.MoveCart(&xcenter, 9,0,60.0,50.0,50.0,-1.0,-1);
51
52
       robot.FT_CalCenterStart();
53
       dir = 1;
54
        osa = 2;
55
       lin_v = 6,0;
56
        maxdis = 150.0;
57
       robot.FT_FindSurface(rcs, dir, axis, lin_v, lin_a, maxdis, ft_goal);
58
       robot.MoveCart(&desc_pos, 9,0,100.0,100.0,100.0,-1.0,-1);
59
       robot.WaitMs(1000);
60
61
       dir = 2;
62
        robot.FT_FindSurface(rcs, dir, axis, lin_v, lin_a, maxdis, ft_goal);
63
                                                                                     (pokračování na další straně)
```

```
64 robot.FT_CalCenterEnd(&ycenter);

printf("ycenter:%f,%f,%f,%f,%f,%f\n",ycenter.tran.x,ycenter.tran.y,ycenter.tran.z,

--ycenter.rpy.rx,ycenter.rpy.ry,ycenter.rpy.rz);

66 robot.MoveCart(&ycenter, 9, 0,60.0,50.0,0.0,-1);

67 vrátit 0;

68 }
```

## 2.1.11.26 Kompatibilní kontrola na

```
1 /**
2 * @brief Kontrola shody na
3 * @param [in] p Koeficient nastavení polohy nebo shody
4 * @param [in] force Práh vyhovující síly otevření, jednotka: N
5 * @return Kód chyby
6 */
6 errno_t FT_ComplianceStart(float p, float force);
```

## 2.1.11.27 Vypnutá kontrola shody

```
1 /**
2 * @brief Vypnutá kontrola shody
3 * @return Kód chyby
4 */
5 errno t FT ComplianceStop();
```

## 2.1.11.28 Příklad kódu

```
#include <cstdlib>
1
   #include <iostream>
2
   #include <stdio.h>
3
   #include <cstring>
4
   #include <unistd.h>
5
   #include "MRobot.h"
6
   #include "RobotTypes.h"
7
8
   using namespace std;
9
10
   int main(void)
11
   Ł
12
                                           //Instituce objektu robota
        MRobot robot:
13
        robot.RPC("192.168.58.2");
                                           /Navázat komunikační spojení s robotem
14
    ↔kontrolér
15
        uint8_t flag = 1;
16
        int sensor_id = 1;
17
        uint8_t select[6] = {1,1,1,0,0,0};
18
                                                                                     (pokračování na další straně)
```

```
(pokračování na předchozí straně)
```

```
float ft_pid[6] = {0.0005,0.0,0.0,0.0,0.0,0.0,0.0};
19
        uint8_t adj_sign = 0;
20
        uint8_t ILC_sign = 0;
21
        float max_dis = 100.0;
22
        float max_ang = 0,0;
23
24
        ForceTorque ft;
25
        DescPose desc_p1, desc_p2, offset_pos;
26
        ExaxisPos epos;
27
        JointPos j1, j2;
28
        memset(&ft, 0, sizeof(ForceTorque));
29
        memset(&desc_p1, 0, sizeof(DescPose));
30
        memset(&desc_p2, 0, sizeof(DescPose));
31
        memset(&offset_pos, 0, sizeof(DescPose));
32
        memset(&j1, 0, sizeof(JointPos));
33
        memset(&j2, 0, sizeof(JointPos));
34
        memset(&epos, 0, sizeof(ExaxisPos));
35
36
        j1 = \{-105.3, -68.0, -127.9, -75.5, 90.8, 77.8\};
37
        j2 = \{-105.3, -97.9, -101.5, -70.3, 90.8, 77.8\};
38
39
        desc_p1.tran.x = -208.9;
40
        desc_p1.tran.y = -274.5;
41
        desc_p1.tran.z = 334.6;
42
        desc_p1.rpy.rx = 178.8;
43
        desc_p1.rpy.ry = -1.3;
44
        desc_p1.rpy.rz = 86,7;
45
46
        desc_p2.tran.x = -264.8;
47
        desc_p2.tran.y = -480.5;
48
        desc_p2.tran.z = 341.8;
49
        desc_p2.rpy.rx = 179.2;
50
        desc_p2.rpy.ry = 0,3;
51
        desc_p2.rpy.rz = 86,7;
52
53
        ft.fx = -10,0;
54
        ft.fy = -10,0;
55
        ft.fz = -10.0;
56
        robot.FT_Control(flag, sensor_id, select, &ft, ft_pid, adj_sign, ILC_sign, max_dis,_
57
     max_ang);
        float p = 0,00005;
58
        float force = 30.0;
59
        robot.FT_ComplianceStart(p, force);
60
        int count = 15;
61
        while (count)
62
        Ł
63
            robot.MoveL(&j1,&desc_p1,9,0,100.0,180.0,100.0,-1.0,&epos,0,1,&offset_pos);
64
            robot.MoveL(&j2,&desc_p2,9,0,100.0,180.0,100.0,-1.0,&epos,0,0,&offset_pos);
65
            count = 1;
66
        }
67
        robot.FT_ComplianceStop();
68
        flag = 0;
69
```

70

71

72 73 }

```
(pokračování na předchozí straně)
```

```
vrátit O;
```

# 2.2 Python

Tato příručka je sekundárním dokumentem vývojového rozhraní jazyka Python.

**Důležité:** Popis jednotky parametrů robota: Jednotkou polohy robota je milimetr (mm) a jednotkou polohy je stupeň (°).

### Důležité:

- 1) V příkladech kódu, které nejsou výslovně uvedeny, byl robot ve výchozím nastavení zapnut a povolen;
- 2) Všechny příklady kódu v dokumentaci jsou ve výchozím nastavení bez zásahu do pracovního prostoru robota;
- 3) Při zkoušce skutečného použití použijte údaje robota na místě.

## 2.2.1 Základní

## 2.2.1.1 Instancování robotů

Prototyp	RPC(ip)
Popis	Instancování objektu robota
Parametr	• ip:IP adresa robota s výchozí tovární IP "192.168.58.2".
Návratová hodnota	<ul> <li>Úspěch: Vrátí objekt robota</li> <li>Neúspěšně: Vytvořený objekt bude zničen</li> </ul>

## 2.2.1.1.1 Příklad kódu

<sup>1</sup> import Mrpc

2

# Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí

<sup>3</sup> robot = Mrpc.RPC('192.168.58.2')

## 2.2.1.2 Číslo verze dotazu SDK

Prototyp	GetSDKVersion()
Popis	Číslo verze dotazu SDK
Parametr	Nic
Návratová hodnota	<ul><li> Úspěch:[0,version]</li><li>Selhání:[errcode,]</li></ul>

### 2.2.1.2.1 Příklad kódu



#### 2.2.1.3 Získání IP adresy řadiče

Prototyp	GetControllerIP()
Popis	Získání IP adresy řadiče
Parametr	Nic
Návratová hodnota	<ul><li>Úspěch:[0,IP]</li><li>Selhání:[errcode,]</li></ul>

#### 2.2.1.3.1 Příklad kódu

```
import Mrpc
1
   # Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí
2
   ↔objekt robota
  robot = Mrpc.RPC('192.168.58.2')
3
   ret = robot.GetControllerIP()
                                      #Získat IP adresu ovladače
4
   it ret[0] == 0:
5
       print("ip řadiče je:",ret[1])
6
   jinak:
7
       print("errcode je: ", ret[0])
8
```

Prototyp	Re <b>ž</b> im(stav)
Popis	Přepínač manuálního/automatického režimu řídicího robota
Parametr	<ul> <li>stav:1-Ruční režim,0-Automatický režim</li> </ul>
Návratová hodnota	<ul><li>Úspěch: [0]</li><li>Selhání:[errcode]</li></ul>

## 2.2.1.4 Přepínač manuálního/automatického režimu řídicího robota

### 2.2.1.4.1 Příklad kódu

1	import	Mrpc
---	--------	------

2

- import time
- <sup>3</sup> *# Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení* vrátí → *objekt robota*
- <sup>4</sup> robot = Mrpc.RPC('192.168.58.2')
- <sup>5</sup> robot.Mode(0) #Robot přejde do automatického provozního režimu
- $_{6}$  time.sleep(1)
- robot.Mode(1) #Robot přejde do manuálního režimu

## 2.2.1.5 Režim přetahování robota

## 2.2.1.5.1 Ovládání robota pro vstup do režimu výuky tažením nebo výstup z něj.

Prototyp	DragTeachSwitch(stav)
Popis	Ovládání robota pro vstup do režimu výuky tažením nebo výstup z něj.
Parametr	• stav:1-Vstup do režimu výuky přetahování,0-Vyjití z režimu výuky přetahování
Návratová hodnota	<ul><li>Úspěch: [0]</li><li>Selhání:[errcode]</li></ul>

## 2.2.1.5.2 Zkontrolujte, zda je robot v režimu přetahování

Prototyp	IsInDragTeach()
Popis	Zkontrolujte, zda je robot v režimu přetahování
Parametr	Nic
Návratová hodnota	<ul> <li>Úspěšnost:[0,state],state:0-Ne drag teaching mode,1-Drag teaching mode</li> <li>Selhání:[errcode]</li> </ul>

## 2.2.1.5.2.1 Příklad kódu

1	import Mrpc		
2	import time		
3	#Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí ⊶objekt robota		
4	robot = Mrpc.RPC('192.168.58.2')		
5	robot.Mode(1) #Robot přejde do ručního režimu		
6	time.sleep(1)		
7	robot.DragTeachSwitch(1) #Když robot přejde do režimu výuky tažením, může pouze _ → vstupte do režimu výuky přetahováním v ručním režimu		
8	time.sleep(1)		
9	ret = robot.IsInDragTeach()#Kontrola, zda je uživatel v režimu přetahování, 1-Režim přetahování $\hookrightarrow Režim_{per}přetahování$ přetahování, 0-		
10	if $ret[0] == 0$ :		
11	print("stav přetažení je:",ret[1])		
12	jinak:		
13	print("errcode je: ", ret[0])		
14	time.sleep(3)		
15	robot.DragTeachSwitch(0) #Když robot přejde do režimu učení bez tahání, může pouze_ → vstupte do režimu výuky bez tahání v ručním režimu		
16	time.sleep(1)		
17	ret = robot.IsInDragTeach()#Kontrola, zda je uživatel v režimu přetahování, 1-Režim $\leftrightarrow$ Žádný režim přetahování)přetahování, 0-		
18	if ret[0] == 0:		
19	print("stav přetažení je:",ret[1])		
20	jinak:		
21	print("errcode je: ", ret[0])		

## 2.2.1.6 Ovládání robota pro povolení nebo snížení povolení

Prototyp	RobotEnable(stav)
Popis	Ovládání robota pro povolení nebo snížení povolení
Parametr	<ul> <li>stav:1-Horní povolení,0-Dolní povolení</li> </ul>
Návratová hodnota	<ul><li>Úspěch: [0]</li><li>Selhání:[errcode]</li></ul>

## 2.2.1.6.1 Příklad kódu

L	import	Mrpc

import time 2

3

#Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí ↔objekt robota

```
4
```

robot = Mrpc.RPC('192.168.58.2') robot.RobotEnable(0) *#Zapnutí robota* 5

(pokračování na další straně)

#### <sup>6</sup> time.sleep(3)

7

robot.RobotEnable(1) *#Tato funkce je na robotovi povolena. Poté, co je* robot \_\_\_\_\_\_ *→ zapnuto, ve výchozím nastavení je automaticky povoleno* 

# 2.2.2 Pohyb

## 2.2.2.1 Robot Jog

## 2.2.2.1.1 jog Jog

Prototyp	StartJOG(ref,nb,dir,vel,acc,max_dis)
Popis	jog Jog
Parametr	<ul> <li>ref:0-kloubový jogging, 2-základní souřadnicový systém jogging, 4-nástrojový souřadnicový systém jogging, 8-obráběcí souřadnicový systém jogging;</li> <li>nb:1-1joint(osa x), 2-2joint(osa y), 3-3join(osa z), 4-4joint(rx), 5-5joint (ry), 6-6joint(rz);</li> <li>dir:0-záporný směr, 1-pozitivní směr;</li> <li>vel:Procento rychlosti,[0~100];</li> <li>acc:Zrychlení v procentech,[0~100];</li> <li>max_dis:Maximální úhel/vzdálenost pro jeden běh, jednotka[° nebo mm]</li> </ul>
Návratová hodnota	<ul><li>Úspěch: [0]</li><li>Selhání:[errcode]</li></ul>

## 2.2.2.1.2 jog jog zpomalení zastaví

Prototyp	StopJOG(ref)
Popis	Jog Jog zpomalem zastavi
Parametr	<ul> <li>ref:1-bodový doraz, 3-bodový doraz souřadnicového systému, 5-bodový doraz souřadnicového systému, 9-bodový doraz souřadnicového systému obrobku</li> </ul>
Návratová hodnota	<ul> <li>Úspěch: [0]</li> <li>Selhání:[errcode]</li> </ul>

## 2.2.2.1.3 jog jog ihned zastaví

Prototyp	ImmStopJOG()
Popis	jog jog se okamžitě zastaví
Parametr	Nic
Návratová hodnota	<ul><li>Úspěch: [0]</li><li>Selhání:[errcode]</li></ul>

## 2.2.2.1.3.1 Příklad kódu

1	import Mrpc	
2	import time	
3	# Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí ↔objekt robota	
4	robot = Mrpc.RPC('192.168.58.2')	
5	# Robot s jedním bodem osy	
6	robot.StartJOG(0,1,0,20.0,20.0,30.0) # Pohyb v jednom kloubu, StartJOG	je ne
	⇔blokovací příkaz a další příkazy pohybu (včetně StartJOG) přijaté během ⇔pohyb bude vyřazen	
7	time.sleep(1)	
8	#Zastavení zpomalení běhu v jedné ose robota	
9	# robot.StopJOG(1)	
10	# Okamžité zastavení oMobot single axis jog	
11	robot.ImmStopJOG()	
12	robot.StartJOG(0,2,1,20.0,20.0,30.0)	
13	time.sleep(1)	
14	robot.ImmStopJOG()	
15	robot.StartJOG(0,3,1,20.0,20.0,30.0)	
16	time.sleep(1)	
17	robot.ImmStopJOG()	
18	robot.StartJOG(0,4,1,20.0,20.0,30.0)	
19	time.sleep(1)	
20	robot.ImmStopJOG()	
21	robot.StartJOG(0,5,1,20.0,20.0,30.0)	
22	time.sleep(1)	
23	[robot.ImmStop]OG()	
24	[rodot.StartJOG(0,6,1,20.0,20.0,30.0)]	
25	time.sieep(1)	
26	robot.ImmStopJOG()	
27	# ZUKIAANI	
28	robot StartIOG(2 1 0 20 0 20 0 100 0) #Logging v $z \dot{\alpha} k ladním sou \dot{r} a dném$	
29		
30	time cloop(1)	
31	#Zastavení zpomalení běhu v jedné ose robota	
32	# robot Ston IOG(2)	
33	# #Immediate stop oMobot single axis iog	
34	robot.ImmStopJOG()	
35	robot.StartJOG(2,1,1,20.0,20.0,100.0)	
36	time.sleep(1)	(nolvněování na dalží stra
	robot.ImmStopJOG()	(pokracovani na uaisi stra

(pokračování na předchozí straně) robot.StartJOG(2,2,1,20.0,20.0,100.0) 37 time.sleep(1) 38 robot.ImmStopJOG() 39 robot.StartJOG(2,3,1,20.0,20.0,100.0) 40 time.sleep(1) 41 robot.ImmStopJOG() 42 robot.StartJOG(2,4,1,20.0,20.0,100.0) 43 time.sleep(1)44 robot.ImmStopJOG() 45 robot.StartJOG(2,5,1,20.0,20.0,100.0) 46 time.sleep(1) 47 robot.ImmStopIOG() 48 robot.StartJOG(2,6,1,20.0,20.0,100.0) 49 time.sleep(1)50 robot.ImmStopJOG() 51 *# Souřadnice nástroje* 52 robot.StartJOG(4,1,0,20.0,20.0,100.0) #Bod v souřadném systému nástroje 53 time.sleep(1) 54 #Zpomalovací stop pro běh v jedné ose robota 55 # robot.StopJOG(5) 56 # #Immediate stop oMobot single axis jog 57 robot.ImmStopJOG() 58 robot.StartJOG(4,1,1,20.0,20.0,100.0) 59 time.sleep(1) 60 robot.ImmStopJOG() 61 robot.StartJOG(4,2,1,20.0,20.0,100.0) 62 time.sleep(1) 63 robot.ImmStopJOG() 64 robot.StartJOG(4,3,1,20.0,20.0,100.0) 65 time.sleep(1) 66 robot.ImmStopJOG() 67 robot.StartJOG(4,4,1,20.0,20.0,100.0) 68 time.sleep(1)69 robot.ImmStopJOG() 70 robot.StartJOG(4,5,1,20.0,20.0,100.0) 71 time.sleep(1) 72 robot.ImmStopJOG() 73 robot.StartJOG(4,6,1,20.0,20.0,100.0) 74 time.sleep(1)75 robot.ImmStopJOG() 76 # Souřadnice práce 77 robot.StartJOG(8,1,0,20.0,20.0,100.0) #Bod v souřadném systému obrobku 78 time.sleep(1)79 #Zpomalovací stop pro běh v jedné ose robota 80 # robot.StopJOG(9) 81 # #Immediate stop oMobot single axis jog 82 robot.ImmStopJOG() 83 robot.StartJOG(8,1,1,20.0,20.0,100.0) 84 time.sleep(1)85 robot.ImmStopJOG() 86 robot.StartJOG(8,2,1,20.0,20.0,100.0) 87

time.sleep(1)

(pokračování na další straně)

89	robot.ImmStopJOG()
90	robot.StartJOG(8,3,1,20.0,20.0,100.0)
91	time.sleep(1)
92	robot.ImmStopJOG()
02	robot.StartĴOG(8,4,1,20.0,20.0,100.0)
55	time.sleep(1)
94	robot.ImmStopJOG()
95	robot.StartJOG(8,5,1,20.0,20.0,100.0)
96	time.sleep(1)
97	robot.ImmStopJOG()
98	robot.StartJOG(8,6,1,20.0,20.0,100.0)
99	time.sleep(1)
100	robot.ImmStopJOG()
101	T - V

#### ----

## 2.2.2.2 Pohyb v kloubním prostoru

MoveJ(joint_pos,desc_pos,tool,user,vel,acc,ovl,exaxis_pos, blendT,offset_flag,offset_pos)
Pohyb v kloubním prostoru
<ul> <li>joint_pos:Cílová poloha kloubu, jednotka[°];</li> <li>desc_pos:Cílová kartézská pozice, jednotka[mm][°];</li> <li>nástroj:Číslo nástroje, [0~14];</li> <li>user:Číslo obrobku, [0~14];</li> <li>vel:Procento rychlosti, [0~100];</li> <li>acc:Zrychlení v procentech, [0~100], dočasně zavřeno;</li> <li>ovl:Faktor škálování rychlosti, [0~100];</li> <li>exaxis_pos:Poloha externí osy 1 až po polohu externí osy 4;</li> <li>blendT:[-1.0]-pohyb na místě (blokovaný), [0-500]-čas vyhlazování (neblokuje se), jednotka[ms];</li> <li>offset_flag:[0]-žádný posun, [1]- posun pod obrobek/základní souřadný systém, [2]-odsazení v souřadnicovém systému nástroje;</li> <li>offset_pos:Posun polohy, jednotka[mm][°]</li> </ul>
<ul><li>Úspěch: [0]</li><li>Selhání:[errcode]</li></ul>

### 2.2.2.2.1 Příklad kódu

1	import Mrpc	
2	import time	
3	# Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí _ ↔objekt robota	
4	robot = Mrpc.RPC('192.168.58.2')	
5	J1=[-168.847,-93.977,-93.118,-80.262,88.985,11.831]	
6	P1=[-558.082,27.343,208.135,-177.205,-0.450,89.288]	
7	eP1=[0.000,0.000,0.000,0.000]	
		(pokračování na další straně)

8	dP1=[1.000,1.000,1.000,1.000,1.000]
9	J2=[168.968,-93.977,-93.118,-80.262,88.986,11.831]
10	P2=[-506.436,236.053,208.133,-177.206,-0.450,67.102]
11	eP2=[0.000,0.000,0.000]
12	dP2=[1.000,1.000,1.000,1.000,1.000]
13	robot.MoveJ(J1, P1, 1, 0, 100.0, 180.0, 100.0, eP1, -1.0, 0, dP1) #Joint space motionPTP, Tool_
	, →číslo1, skutečná zkouška vychází z údajů z terénu a čísla nástroje
14	robot.MoveJ(J2,P2,1,0,100.0,180.0,100.0,eP2,-1.0,0,dP2)
15	time.sleep(2)
16	j1 = robot.GetInverseKin(0,P1,-1) #Pouze v případě kartézských souřadnic, → k řešení polohy kloubu lze použít inverzní kinematické rozhraní.
17	print(j1)
18	j1 = [j1[1], j1[2], j1[3], j1[4], j1[5], j1[6]]
19	robot.MoveJ(j1,P1,1,0,100.0,180.0,100.0,eP1,-1.0,0,dP1)
20	j2 = robot.GetInverseKin(0,P2,-1)
21	print(j2)
22	j2 = [j2[1], j2[2], j2[3], j2[4], j2[5], j2[6]]
23	robot.MoveJ(j2,P2,1,0,100.0,180.0,100.0,eP2,-1.0,0,dP2)
24	time.sleep(2)
25	p1 = robot.GetForwardKin(J1) #K řešení lze použít dopředné kinematické rozhraní →Souřadnice kartezského prostoru pouze s polohami kloubů
26	print(p1)
27	p1 = [p1[1], p1[2], p1[3], p1[4], p1[5], p1[6]]
28	robot.MoveJ(J1,p1,1,0,100.0,180.0,100.0,eP1,-1.0,0,dP1)
29	p2 = robot.GetForwardKin(J2)
30	print(p2)
31	p2 = [p2[1], p2[2], p2[3], p2[4], p2[5], p2[6]]
32	robot.MoveJ(J2,p2,1,0,100.0,180.0,100.0,eP2,-1.0,0,dP2)

## 2.2.2.3 Lineární pohyb v kartézském prostoru

Prototyp	MoveL(joint_pos,desc_pos,tool,user,vel,acc,ovl,blendR, exaxis_pos,search.offset_flag.offset_pos)
Popis	Lineární pohyb v kartézském prostoru
Parametr	<ul> <li>joint_pos:Cílová poloha kloubu, jednotka[°];</li> <li>desc_pos:Cílová kartézská pozice, jednotka[mm][°];</li> <li>nástroj:Číslo nástroje, [0~14];</li> <li>user:Číslo obrobku, [0~14];</li> <li>vel:Procento rychlosti, [0~100];</li> <li>acc:Zrychlení v procentech, [0~100], dočasně zavřeno;</li> <li>ovl:Faktor škálování rychlosti, [0~100];</li> <li>blendR:[-1.0]-pohyb na místě (blokovaný), [0-1000]-plynulý poloměr(neblokuje), jednotka[mm];</li> <li>exaxis_pos:Poloha externí osy 1~poloha externí osy 4;</li> <li>vyhledávání: [0]- umístění nesvařovacího drátu, [1]- umístění svařovacího drátu;</li> <li>offset_flag:[0]-žádný offset, [1]-offset v souřadném systému obrobku/základny, [2]-offset v souřadném systému nástroje;</li> <li>offset_pos:Posun polohy jednotka[mm][°]</li> </ul>
Návratová hodnota	<ul> <li>Úspěch: [0]</li> <li>Selhání:[errcode]</li> </ul>

# 2.2.2.3.1 Příklad kódu

1	import Mrpc
2	#Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí∟ ⊶objekt robota
3	robot = Mrpc.RPC('192.168.58.2')
4	J1=[95.442,-101.149,-98.699,-68.347,90.580,-47.174]
5	P1=[75.414,568.526,338.135,-178.348,-0.930,52.611]
6	eP1=[0.000,0.000,0.000]
7	dP1=[10.000,10.000,10.000,0.000,0.000]
8	J2=[123.709,-121.190,-82.838,-63.499,90.471,-47.174]
9	P2=[-273.856,643.260,259.235,-177.972,-1.494,80.866]
10	eP2=[0.000, 0.000, 0.000, 0.000]
11	dP2=[0.000, 0.000, 0.000, 0.000, 0.000]
12	$J_{3}=[16/.066, -95.700, -123.494, -42.493, 90.466, -47.174]$
13	$P_3 = [-423.044, 229.703, 241.080, -173.990, -5.772, 123.971]$
14	$P_{2}=[0.000, 0.000, 0.000]$
15	
16	robot.MoveL(J1,P1,0,0,100.0,180.0,100.0,-1.0,eP1,0,1,dP1) #Rectilinearni pohyb V
10	$\rightarrow Karrezsky prosion$
47	10D0t.MOVeL(J2,P2,0,0,100.0,180.0,100.0,-1.0,eP2,0,0,dP2)
1/	robot.MoveL(J3,P3,0,0,100.0,180.0,100.0,-1.0,eP3,0,0,dP3)
18	

#### 2.2.2.4 Pohyb po kruhovém oblouku v kartézském prostoru

Prototyp	MoveC(joint_pos_p,desc_pos_p,ptool,puser,pvel,pacc,exaxis_pos_p, poffset_flag,offset_pos_p,joint_pos_t,desc_pos_t,ttool,tuser, tvel,tacc,exaxis_pos_t,toffset_flag,offset_pos_t,ovl,blendR)
Popis	Pohyb po kruhovém oblouku v kartézském prostoru
Parametr	<ul> <li>joint_pos_p:Poloha kloubu bodu cesty, jednotka[°];</li> <li>desc_pos_p:Kartézská pozice bodu cesty, jednotka[mm][°];</li> <li>ptool:Číslo nástroje,[0~14];</li> <li>puser:Číslo obrobku,[0~14];</li> <li>pvel:Procento rychlosti,[0~100];</li> <li>pacc:Zrychlení v procentech,[0~100],dočasně zavřeno;</li> <li>exaxis_pos_p:Poloha externí osy 1~poloha externí osy 4;</li> <li>poffset_flag:[0]-žádný posun, [1]- posun v souřadném systému obrobku/základny, [2]- posun v souřadném systému nástroje;</li> <li>offset_pos_p:Offset,jednotka[mm][°];</li> <li>joint_pos_t:Poloha cílového bodu kloubu, jednotka[°];</li> <li>desc_pos_t:kartézská pozice cílového bodu, jednotka[mm][°];</li> <li>ttool:Číslo nástroje,[0~14];</li> <li>tuser:Číslo obrobku,[0~14];</li> <li>tvel:Rychlost v procentech,[0~100];</li> <li>tacc: procento zrychlení,[0~100], dočasně zavřeno;</li> <li>exaxis_pos_t:Poloha vnější osy 1~poloha vnější osy 4;</li> <li>toffset_flag:[0]-žádný posun, [1]- posun v souřadném systému obrobku/základny, [2]- posun v souřadném systému nástroje;</li> <li>offset_flag:[0]-žádný posun, [1]- posun v souřadném systému obrobku/základny, [2]- posun v souřadném systému nástroje;</li> <li>offset_flag:[0]-žádný posun, [1]- posun v souřadném systému obrobku/základny, [2]- posun v souřadném systému nástroje;</li> <li>offset_pos_t:Offset,unit[mm][°]</li> <li>ovl:Faktor škálování rychlosti,[0~100];</li> <li>blendR:[-1.0]-pohyb na místě (blokovaný), [0-1000]-plynulý poloměr(neblokovaný),jednotka[mm]</li> </ul>
Návratová hodnota	<ul><li>Úspěch: [0]</li><li>Selhání:[errcode]</li></ul>

## 2.2.2.4.1 Příklad kódu

```
import Mrpc
1
   # Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí
2
   ↔objekt roboła
   robot = Mrpc.RPC('192.168.58.2')
3
   J1=[121.381,-97.108,-123.768,-45.824,89.877,-47.296]
4
   P1=[-127.772,459.534,221.274,-177.850,-2.507,78.627]
5
   eP1=[0.000,0.000,0.000,0.000]
6
   dP1=[10.000,10.000,10.000,10.000,10.000]
7
   J2=[138.884,-114.522,-103.933,-49.694,90.688,-47.291]
8
   P2=[-360.468,485.600,196.363,-178.239,-0.893,96.172]
9
   eP2=[0.000,0.000,0.000,0.000]
10
   dP2=[10.000,10.000,10.000,10.000,10.000]
11
```

```
pa2=[0.0,0.0,100.0,180.0]
```

(pokračování na další straně)

13	J3=[159.164,-96.105,-128.653,-41.170,90.704,-47.290]	
14	P3=[-360.303,274.911,203.968,-176.720,-2.514,116.407]	
15	eP3=[0.000,0.000,0.000,0.000]	
16	dP3=[10.000,10.000,10.000,10.000,10.000]	
17	pa3=[0.0,0.0,100.0,180.0]	
18	dP=[10.000,10.000,10.000,10.000,10.000]	
19	robot.MoveJ(J1,P1,0,0,100.0,180.0,100.0,eP1,-1.0,0,dP1)	#Společný pohyb v
20	robot.MoveC(J2,P2,pa2,eP2,0,dP2,J3,P3,pa3,eP3,0,dP3,100.0,-1.0)	prostou PTP Krithový pohyb V
	↔Kartézský prostor	

## 2.2.2.5 Kruhový pohyb v kartézském prostoru

Prototyp	Circle(joint_pos_p,desc_pos_p,ptool,puser,pvel,pacc, exaxis_pos_p,joint_pos_t,desc_pos_t,ttool,tuser,tvel,tacc, exaxis_pos_t,ovl,offset_flag,offset_pos)
Popis	Kruhový pohyb v kartézském prostoru
Parametr	<ul> <li>joint_pos_p:Poloha kloubu bodu cesty, jednotka[°];</li> <li>desc_pos_p:Kartézská pozice bodu cesty, jednotka[mm][°];</li> <li>ptool:Číslo nástroje,[0~14];</li> <li>puser:Číslo obrobku,[0~14];</li> <li>pvel:Procento rychlosti,[0~100];</li> <li>pacc:Zrychlení v procentech,[0~100],dočasně zavřeno;</li> <li>exaxis_pos_p:Poloha externí osy 1~poloha externí osy 4;</li> <li>joint_pos_t:Poloha cílového bodu kloubu, jednotka[°];</li> <li>desc_pos_t:kartézská pozice cílového bodu, jednotka["];</li> <li>ttool:Číslo nástroje,[0~14];</li> <li>tuser:Číslo obrobku,[0~14];</li> <li>tuser:Číslo obrobku,[0~14];</li> <li>tuser:Číslo obrobku,[0~14];</li> <li>tuser:Číslo obrobku,[0~14];</li> <li>tvel:Rychlost v procentech,[0~100];</li> <li>tacc: procento zrychlení,[0~100], dočasně zavřeno;</li> <li>exaxis_pos_t:Poloha vnější osy 1~poloha vnější osy 4;</li> <li>ovl:Faktor škálování rychlosti,[0~100%];</li> <li>offset_flag:[0]-žádný offset, [1]-offset v souřadném systému nástroje;</li> <li>offset_pos:Offset,jednotka[mm][°]</li> </ul>
Návratová hodnota	<ul><li>Úspěch: [0]</li><li>Selhání:[errcode]</li></ul>

## 2.2.2.5.1 Příklad kódu

1	import Mrpc
2	# Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí ↔objekt robota
3	robot = Mrpc.RPC('192.168.58.2')
4	J1=[121.381,-97.108,-123.768,-45.824,89.877,-47.296]
5	P1=[-127.772,459.534,221.274,-177.850,-2.507,78.627]
6	eP1=[0.000,0.000,0.000]
7	dP1=[10.000,10.000,10.000,10.000,10.000]
8	J2=[138.884,-114.522,-103.933,-49.694,90.688,-47.291]
9	P2=[-360.468,485.600,196.363,-178.239,-0.893,96.172]
10	eP2=[0.000,0.000,0.000]
11	dP2=[10.000, 10.000, 10.000, 10.000, 10.000]
12	pa2=[0.0, 0.0, 100.0, 180.0]
13	$J_{2} = [159.104, -90.105, -128.055, -41.170, 90.704, -47.290]$
14	$P_{2} = [-500.505, 274.911, 205.908, -170.720, -2.514, 110.407]$ $a_{2} = [0, 000, 0, 000, 0, 000, 0, 000]$
15	dP3 = [10,000,0.000,0.000,0.000]
16	$n_{2} = [0.000, 10.000, 10.000, 10.000, 10.000]$
17	dP = [10, 000, 10, 000, 10, 000, 10, 000, 10, 000]
18	
19	robot.MoveJ(J1,P1,0,0,100.0,180.0,100.0,eP1,-1.0,0,dP1) #Pohyb v kloubovém prostoruPTP
20	robot.Circle(J2,P2,pa2,eP2,J3,P3,pa3,eP3,100.0,0,dP) #Kruhový pohyb v kartézské
	soustavě
	' <i>↔prostor</i>

## 2.2.2.6 Spirálový pohyb v kartézském prostoru

		-
Prototyp	NewSpiral(joint_pos,desc_pos,tool,user,vel,acc,exaxis_pos,ovl, offset_flag,offset_pos,param)	
Popis	Spirálový pohyb v kartézském prostoru	
Parametr	<ul> <li>joint_pos:Cílová poloha kloubu, jednotka[°];</li> <li>desc_pos:Cílová kartézská pozice, jednotka[mm][°];</li> <li>nástroj:Číslo nástroje, [0~14];</li> <li>user:Číslo obrobku, [0~14];</li> <li>vel:Procento rychlosti, [0~100], dočasně zavřeno;</li> <li>exaxis_pos:Poloha externí osy 1~poloha externí osy 4;</li> <li>ovl:Faktor škálování rychlosti, [0~100];</li> <li>offset_flag:[0]-žádný offset, [1]-offset v souřadném systému obrobku/základny, [2]-offset v souřadném systému nástroje;</li> <li>offset_pos:Posun polohy, jednotka[mm][°]</li> <li>param:[circle_num,circle_angle,rad_init,rad_add,rotaxis_add,rot_direction],circl počet závitů, circle_angle: úhel šroubovice, rad_init: počáteční poloměr šroubovice, rad_add: přírůstek poloměru, rota_add: přírůstek směru osy, rot_direction: směr otáčení, 0-po směru hodinových ručiček, 1-proti směru hodinových ručiček</li> </ul>	e_num:
Návratová hodnota	<ul> <li>Úspěch: [0]</li> <li>Selhání:[errcode]</li> </ul>	

### 2.2.2.6.1 Příklad kódu

1	import Mrpc
2	#Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení <mark>vrátí</mark> ⊶objekt robota
3	robot = Mrpc.RPC('192.168.58.2')
4	J1=[127.888,-101.535,-94.860,17.836,96.931,-61.325]
5	eP1=[0.000,0.000,0.000]
6	dP1=[50.0,0.0,0.0,-30.0,0.0,0.0]
7	]2=[127.888,-101.535,-94.860,17.836,96.931,-61.325]
8	eP2=[0.000, 0.000, 0.000]
9	$ \frac{dP2}{D0} = \begin{bmatrix} 5 & 0 & 5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0$
10	Pa = [5.0, 5.0, 50.0, 10.0, 10.0, 0.0]
11	$P1 = robot.GetForWardKin(J1)$ #K resent ize pouzit dopredne kinematicke roznrani_ $\Rightarrow Souřadnice kanematicke prostoru pouze s polohami kloubů$
	print(P1)
12	P1 = [P1[1], P1[2], P1[3], P1[4], P1[5], P1[6]]
13	robot.MoveJ(J1,P1,0,0,100.0,180.0,100.0,eP1,0.0,2,dP1)
14	P2 = robot GetForwardKin(I2) #K řešení lze použít dopředné kinematické rozhraní
15	Souřadnice kartézského prostoru pouze s polohami kloubů
16	print(P2)
17	P2 = [P2[1], P2[2], P2[3], P2[4], P2[5], P2[6]]
18	robot.NewSpiral(J2,P2,0,0,100.0,180.0,eP2,100.0,2,dP2,Pa)

#### 2.2.2.7 Pohyb v kloubním prostoru v režimu serva

Prototyp Popis	ServoJ(joint_pos,acc,vel,cmdT,filterT,gain) Pohyb v kloubním prostoru v režimu serva
Parametr	<ul> <li>joint_pos:Cílová poloha kloubu, jednotka[°];</li> <li>acc:Zrychlení, rozsah[0~100], dočasně zavřeno, výchozí hodnota 0;</li> <li>vel: Rychlost, rozsah [0~100], dočasně uzavřeno, výchozí hodnota 0;</li> <li>cmdT:Instruction Cycle,unit[s],[0.001~0.016];</li> <li>filterT:Čas filtrování,jednotka[s],dočasně uzavřeno;</li> <li>zesílení: Proporcionální zesilovač pro cílovou polohu, dočasně uzavřený</li> </ul>
Návratová hodnota	<ul><li>Úspěch: [0]</li><li>Selhání:[errcode]</li></ul>

### 2.2.2.7.1 Příklad kódu

- 1 import Mrpc
- 2 import time

3

- # Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí ↔objekt robota
- 4 robot = Mrpc.RPC('192.168.58.2')
- 5 joint\_pos = robot.GetActualJointPosDegree(0)

(pokračování na další straně)

```
print(joint_pos)
6
   joint_pos = [ joint_pos[1], joint_pos[2], joint_pos[3], joint_pos[4], joint_pos[5], joint_
7
    ,→pos[6]]
   acc = 0,0
8
   vel = 0.0
9
   t = 0.008
10
   lookahead_time = 0.0
11
   P = 0.0
12
   count = 100
13
   while(count):
14
        robot.ServoJ(joint_pos, acc, vel, t, lookahead_time, P)
15
        joint_pos[0] = joint_pos[0] + 0.1
16
        count = count - 1
17
        time.sleep(0.008)
18
```

### 2.2.2.8 Pohyb v režimu serva v kartézském prostoru

Prototyp	ServoCart(mode,desc_pos,pos_gain,acc,vel,cmdT,filterT,gain)
Popis	Pohyb v režimu serva v kartézském prostoru
Parametr	<ul> <li>režim: [0]-absolutní pohyb (základní souřadný systém), [1]-inkrementální pohyb (základní souřadný systém), [2]-inkrementální pohyb (souřadný systém nástroje);</li> <li>desc_pos:Cílová kartézská poloha / přírůstek cílové kartézské polohy;</li> <li>pos_gain:Koeficient přírůstku polohy, účinný pouze při inkrementální motizaci, rozsah[0~1];</li> <li>acc:Zrychlení, rozsah[0~100], dočasně zavřeno, výchozí hodnota 0;</li> <li>vel: Rychlost, rozsah [0~100], dočasně uzavřeno, výchozí hodnota 0;</li> <li>cmdT:Instruction Cycle,unit[s],[0.001~0.016];</li> <li>filterT:Čas filtrování,jednotka[s],dočasně uzavřeno;</li> <li>zesílení: Proporcionální zesilovač pro cílovou polohu, dočasně uzavřený</li> </ul>
Návratová hodnota	<ul> <li>Úspěch: [0]</li> <li>Selhání:[errcode]</li> </ul>

## 2.2.2.8.1 Příklad kódu

```
import Mrpc
1
   import time
2
   # Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí
3
    ↔objekt robota
   robot = Mrpc.RPC('192.168.58.2')
4
   mode = 2 #Přírůstkový pohyb v souřadném systému nástroje
5
   n_{pos} = [0.0, 0.0, 0.5, 0.0, 0.0, 0.0, 0.0]
                                           #Inkrementální pozice v kartézském prostoru
6
   gain = [0.0,0.0,1.0,0.0,0.0,0.0,0.0]
7
   acc = 0,0
8
   vel = 0,0
9
10
   t = 0.008
                                                                                      (pokračování na další straně)
```

```
11 lookahead_time = 0.0
12 P = 0.0
13 count = 100
14 while(count):
15 robot.ServoCart(mode, n_pos, gain, acc, vel, t, lookahead_time, P)
16 count = count - 1
17 time.sleep(0.008)
```

#### 2.2.2.9 Pohyb z bodu do bodu v kartézském prostoru

Prototyp	MoveCart(desc_pos,tool,user,vel,acc,ovl,blendT,config)
Popis	Pohyb z bodu do bodu v kartézském prostoru
Parametr	<ul> <li>desc_pos:Cílová kartézská poloha;</li> <li>nástroj:Číslo nástroje,[0~14];</li> <li>user:Číslo obrobku,[0~14];</li> <li>vel: Rychlost, rozsah [0~100], dočasně uzavřeno, výchozí hodnota 0;</li> <li>acc:Zrychlení, rozsah[0~100], dočasně zavřeno, výchozí hodnota 0;</li> <li>ovl:Faktor škálování rychlosti,[0~100];</li> <li>blendT:[-1.0]-pohyb na místě (blokovaný), [0-500]-čas vyhlazování (neblokuje se),jednotka[ms];</li> <li>config:Joint configuration, [-1]-odkazuje na aktuální pozici kloubu pro řešení, [0-7]-řešit na základě společné konfigurace</li> </ul>
Návratová hodnota	<ul><li>Úspěch: [0]</li><li>Selhání:[errcode]</li></ul>

## 2.2.2.9.1 Příklad kódu

1	import Mrpc
2	import time
3	#Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí ↔objekt robota
4	robot = Mrpc.RPC('192.168.58.2')
5	P1=[75.414,568.526,338.135,-178.348,-0.930,52.611]
6	P2=[-273.856,643.260,259.235,-177.972,-1.494,80.866]
7	P3=[-423.044,229./03,241.080,-1/3.990,-5.//2,123.9/1]
8	robot.MoveCart(P1,0,0,100.0,100.0,100.0,-1.0,-1) #Point-to-point motion in $\checkmark$ Kartézský prostor
9	robot.MoveCart(P2,0,0,100.0,100.0,100.0,-1.0,-1)
10	robot.MoveCart(P3,0,0,100.0,100.0,100.0,0.0,-1)
11	time.sleep(1)
12	robot.StopMotion() #Zastavení
	pohybu

## 2.2.2.10 Robot spline motion

## 2.2.2.10.1 Spline motion start

Prototyp	SplineStart()
Popis	Spline motion start
Parametr	Nic
Návratová hodnota	<ul><li>Úspěch: [0]</li><li>Selhání:[errcode]</li></ul>

## 2.2.2.10.2 Drážkový pohyb PTP

Prototyp Popis Parametr	SplinePTP(joint_pos,desc_pos,tool,user,vel,acc,ovl) Drážkový pohyb PTP
	<ul> <li>joint_pos:Cílová poloha kloubu, jednotka[°];</li> <li>desc_pos:Cílová kartézská pozice, jednotka[mm][°];</li> <li>nástroj:Číslo nástroje, [0~14];</li> <li>uživatel:Číslo obrobku, [0~14];</li> <li>vel: Rychlost, rozsah [0~100], dočasně uzavřeno, výchozí hodnota 0;</li> <li>acc:Zrychlení, rozsah[0~100], dočasně zavřeno, výchozí hodnota 0;</li> <li>ovl:Faktor škálování rychlosti, [0~100];</li> </ul>
Návratová hodnota	<ul><li>Úspěch: [0]</li><li>Selhání:[errcode]</li></ul>

## 2.2.2.10.3 Konec pohybu drážkování

Prototyp	SplineEnd()
Popis	Konec pohybu drážkování
Parametr	Nic
Návratová hodnota	<ul> <li>Úspěch: [0]</li> <li>Selhání:[errcode]</li> </ul>

## 2.2.2.10.3.1 Příklad kódu

1	(import Mrpc
2	#Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí ↔objekt robota
3	robot = Mrpc.RPC('192.168.58.2')
4	J1 = [114.578,-117.798,-97.745,-54.436,90.053,-45.216]
5	P1 = [-140.418,619.351,198.369,-179.948,0.023,69.793]
6	eP1 = [0.000,0.000,0.000,0.000]
7	dP1 = [0,000,0,000,0,000,0,000,0,000,0,000]
8	J2 = [115.401,-105.206,-117.959,-49.727,90.054,-45.222]
9	P2 = [-95.586,504.143,186.880,178.001,2.091,70.585]
10	J3 = [135.609,-103.249,-120.211,-49.715,90.058,-45.219]
11	P3 = [-252.429,428.903,188.492,177.804,2.294,90.782]
12	J4 = [154.766,-87.036,-135.672,-49.045,90.739,-45.223]
13	P4 = [-277.255,272.958,205.452,179.289,1.765,109.966]
14	robot.MoveJ(J1,P1,0,0,100.0,180.0,100.0,eP1,-1.0,0,dP1)
15	robot.SplineStart() #Spline motion start
16	robot.SplinePTP(J1,P1,0,0,100.0,180.0,100.0) # <i>Spline motion PTP</i>
17	robot.SplinePTP(J2,P2,0,0,100.0,180.0,100.0)
18	robot.SplinePTP(J3,P3,0,0,100.0,180.0,100.0)
19	robot.SplinePTP(J4,P4,0,0,100.0,180.0,100.0)
20	robot.SplineEnd() #Konec spline pohybu

## 2.2.2.11 Robot New Spline Motion

## 2.2.2.11.1 Nový začátek pohybu spline

Prototyp	NewSplineStart(type)
Popis	Nový začátek pohybu spline
Parametr	<ul> <li>typ:0-obloukový přechod, 1-daný bod pozice bod cesty</li> </ul>
Návratová hodnota	<ul><li>Úspěch: [0]</li><li>Selhání:[errcode]</li></ul>

## 2.2.2.11.2 Nové drážkové zakončení pohybu

Prototyp	NewSplineEnd()
Popis	Nové drážkové zakončení pohybu
Parametr	Nic
Návratová hodnota	<ul><li>Úspěch: [0]</li><li>Selhání:[errcode]</li></ul>

## 2.2.2.11.3 Nové body pokynů Spline

Prototyp Popis	NewSplinePoint(joint_pos,desc_pos,tool,user,vel,acc,ovl,blendR, lastFlag) Nové body pokynů Spline
Parametr	<ul> <li>joint_pos:Cílová poloha kloubu, jednotka[°];</li> <li>desc_pos:Cílová kartézská pozice, jednotka[mm][°];</li> <li>nástroj:Číslo nástroje, [0~14];</li> <li>uživatel:Číslo obrobku, [0~14];</li> <li>vel: Rychlost, rozsah [0~100], dočasně uzavřeno, výchozí hodnota 0;</li> <li>acc:Zrychlení, rozsah[0~100], dočasně zavřeno, výchozí hodnota 0;</li> <li>ovl:Faktor škálování rychlosti, [0~100];</li> <li>blendR: [0-1000]-hladký poloměr, jednotka[mm];</li> <li>lastFlag:Je to poslední bod, 0-Ne, 1-Ano</li> </ul>
Návratová hodnota	<ul> <li>Úspěch: [0]</li> <li>Selhání:[errcode]</li> </ul>

# 2.2.2.11.3.1 Příklad kódu

1	(import Mrpc
2	# Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí ↔objekt robota
3	robot = Mrpc.RPC('192.168.58.2')
4	J1 = [114.578,-117.798,-97.745,-54.436,90.053,-45.216]
5	P1 = [-140.418,619.351,198.369,-179.948,0.023,69.793]
6	eP1 = [0.000,0.000,0.000,0.000]
7	dP1 = [0,000,0,000,0,000,0,000,0,000,0,000]
8	J2 = [115.401,-105.206,-117.959,-49.727,90.054,-45.222]
9	P2 = [-95.586,504.143,186.880,178.001,2.091,70.585]
10	J3 = [135.609,-103.249,-120.211,-49.715,90.058,-45.219]
11	P3 = [-252.429,428.903,188.492,177.804,2.294,90.782]
12	J4 = [154.766,-87.036,-135.672,-49.045,90.739,-45.223]
13	P4 = [-277.255,272.958,205.452,179.289,1.765,109.966]
14	robot.MoveJ(J1,P1,0,0,100.0,180.0,100.0,eP1,-1.0,0,dP1)
15	robot.NewSplineStart(1) #Začíná pohyb spline
16	robot.NewSplinePoint(J1,P1,0,0,50.0,50.0,50.0,0.0,0) # <i>Řídicí bod</i>
17	robot.NewSplinePoint(J2,P2,0,0,0,50.0,50.0,50.0,0.0,0)
18	robot.NewSplinePoint(J3,P3,0,0,0,50.0,50.0,50.0,0.0,0)
19	robot.NewSplinePoint( $J4, P4, 0, 0, 0, 50.0, 50.0, 50.0, 0, 0, 1$ )
20	robot.newspineEnd()

## 2.2.2.12 Robot ukončí pohyb

Prototyp	StopMotion()
Popis	Chcete-li ukončit pohyb, použijte instrukce pro ukončení pohybu jako neblokující stav.
Parametr	Nic
Návratová hodnota	<ul><li>Úspěch: [0]</li><li>Selhání:[errcode]</li></ul>

### 2.2.2.12.1 Příklad kódu

1	import Mrpc
2	import time
3	# Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí ∟ ↔objekt robota
4	robot = Mrpc.RPC('192.168.58.2')
5	P1=[75.414,568.526,338.135,-178.348,-0.930,52.611]
6	P2=[-273.856,643.260,259.235,-177.972,-1.494,80.866]
7	P3=[-423.044,229./03,241.080,-1/3.990,-5.//2,123.9/1]
8	robot.MoveCart(P1,0,0,100.0,100.0,100.0,-1.0,-1)#Pohyb z bodu do bodu v kloubu_
9	robot.MoveCart(P2,0,0,100.0,100.0,100.0,-1.0,-1)
10	robot.MoveCart(P3,0,0,100.0,100.0,100.0,0.0,-1) <i>#Tento pokyn k pohybu je v ne</i> <i>↔blokovací stav</i>
11	time.sleep(1)
12	robot.StopMotion() #Zastavení
	pohybu

## 2.2.2.13 Celkový posun oMobot bodů

## 2.2.2.13.1 Výchozí bod celkový posun

Prototyp	PointsOffsetEnable(flag,offset_pos)
Popis	výchozí bod čelkový posuh
Parametr	<ul> <li>příznak:0-posun v základním souřadnicovém systému nebo v souřadnicovém systému obrobku, 2-posun v souřadnicovém systému nástroje;</li> <li>offset_pos:Offset,jednotka[mm][°]</li> </ul>
Návratová hodnota	<ul><li>Úspěch: [0]</li><li>Selhání:[errcode]</li></ul>

## 2.2.2.13.2 Celkový posun konců bodu

Prototyp	PointsOffsetDisable()
Popis	Celkový posun konců bodu
Parametr	Nic
Návratová hodnota	<ul><li>Úspěch: [0]</li><li>Selhání:[errcode]</li></ul>

## 2.2.2.13.2.1 Příklad kódu

1	import Mrpc
2	import time
3	#Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí ⊶objekt robota
4 5	robot = Mrpc.RPC('192.168.58.2') #Celkový posun pozice bodu oMobot
6	J1=[-168.84/,-93.9//,-93.118,-80.262,88.985,11.831]
7	P1=[-558.082,27.343,208.135,-177.205,-0.450,89.288]
8	ePI=[0.000,0.000,0.000,0.000]
9	dP1=[10.000, 10.000, 10.000, 0.000, 0.000]
10	JZ = [108.908, -93.977, -93.118, -80.202, 88.980, 11.831]
11	P2-[-500.450,250.055,208.155,-177.200,-0.450,07.102]
12	dP2 = [0.000, 0.000, 0.000, 0.000]
13	robot. MoveI(I1, P1, 1, 0, 100, 0, 180, 0, 100, 0, eP1, -1, 0, 0, dP1)
14	robot.MoveI(I2, P2, 1, 0, 100, 0, 180, 0, 100, 0, eP2, -1, 0, 0, dP2)
15	time.sleep(2)
16	flag = 0
17	offset = $[100, 0.5, 0.6, 0.0, 0, 0.0, 0, 0, 0]$ #Pose offset
18	
19	robot.PointsOffsetEnable(flag, offset) #Globalni pocatecni posun
20 21	robot.MoveJ(J2,P2,1,0,100.0,180.0,100.0,eP2,-1.0,0,dP2)
22	robot.PointsOffsetDisable() #Konec globálního posunu

# 2.2.3 IO

## 2.2.3.1 Nastavení digitálního výstupu řídicí jednotky

Prototyp Popis Parametr	SetDO(id,status,smooth,block) Nastavení digitálního výstupu řídicí jednotky • id:číslo IO,rozsah[0~15]; • stav:0-off, 1-on; • hladký:0-nehladký, 1-hladký; • blokování:0-blokování, 1-neblokování.
Návratová hodnota	<ul> <li>úspěch:[0]</li> <li>Selhání:[errcode]</li> </ul>

## 2.2.3.1.1 Příklad kódu

1	(import Mrpc
2	# Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí ↔objekt robota
3	robot = Mrpc.RPC('192.168.58.2')
4	for i in range(0,16):
5	robot.SetDO(i,1,0,0) <i>#Otevřete ovládací pole DO</i>
6	robot.WaitMs(1000)
7	for i in range(0,16):
8	robot.SetDO(i,0,0,0) #Zavřete ovládací pole DO
9	robot.WaitMs(1000)

## 2.2.3.2 Nastavení digitálního výstupu nástroje

Prototyp Popis	SetToolDO(id,status,smooth,block) Nastavení digitálního výstupu nástroje
Parametr	<ul> <li>id:číslo IO,rozsah[0~15];</li> <li>stav:0-off, 1-on;</li> <li>hladký:0-nehladký, 1-hladký;</li> <li>blokování:0-blokování, 1-neblokování.</li> </ul>
Návratová hodnota	<ul> <li>Úspěch: [0]</li> <li>Selhání:[errcode]</li> </ul>

## 2.2.3.2.1 Příklad kódu

```
import Mrpc
1
   #Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí
↔objekt robota
2
   robot = Mrpc.RPC('192.168.58.2')
3
   for i in range(0,2):
robot.SetToolDO(i,1,0,0)
robot.WaitMs(1000)
4
5
                                              #Otevřete ovládací pole DO
6
   robot.WaitMs(1000)
7
   for i in range(0,2):
robot.SetTooIDO(1,0,0,0)
8
                                              #Zavrete ovladaci pole DO
9
```

#### 2.2.3.3 Nastavení analogového výstupu ovládacího pole

Prototyp Popis	SetAO(id,value,block) Nastavení analogového výstupu řídicí jednotky
Parametr	<ul> <li>id:číslo IO,rozsah[0~1];</li> <li>hodnota:elektrická energie nebo napětí v procentech, rozsah [0-100%] odpovídá elektrické energii [0-20mA] nebo napětí [0-10V];</li> <li>blok: [0]- blokování, [1]- neblokující</li> </ul>
Návratová hodnota	<ul><li>Úspěch: [0]</li><li>Selhání:[errcode]</li></ul>

## 2.2.3.3.1 Příklad kódu

1	import Mrpc
2	#Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení <mark>vrátí</mark> ↔objekt robota
3	robot = Mrpc.RPC('192.168.58.2')
4	robot.SetAO(0,0.0,0) # Nastavení analogového výstupu řídicího boxu
5	robot.WaitMs(1000)
6	robot.SetAO(1,100.0,0)

## 2.2.3.4 Nastavení analogového výstupu nástroje

Prototyp Popis	SetToolAO(id,value,block) Nastavení analogového výstupu nástroje
Parametr	<ul> <li>id:IO number,range[0];</li> <li>hodnota:elektrická energie nebo napětí v procentech, rozsah [0-100%] odpovídá elektrické energii [0-20mA] nebo napětí [0-10V];</li> <li>blok: [0]- blokování, [1]- neblokující</li> </ul>
Návratová hodnota	<ul><li>Úspěch: [0]</li><li>Selhání:[errcode]</li></ul>

#### 2.2.3.4.1 Příklad kódu

1 2

(import Mrpc
# Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí ↔objekt robota
robot = Mrpc.RPC('192.168.58.2')
robot.SetToolAO(0,100.0,0) # Nastavení analogového výstupu nástroje
robot. WaitMs(1000)
robot.SetToolAO(0,0.0,0)

## 2.2.3.5 Získejte digitální vstup řídicí jednotky

Prototyp	GetDI(id,block)
Popis	Získejte digitální vstup řídicí jednotky
Parametr	<ul> <li>id:číslo IO,rozsah[0~15];</li> <li>blok: [0]- blokování, [1]- neblokující</li> </ul>
Návratová hodnota	<ul> <li>Úspěch:[0,di],di: 0-Nízká úroveň,1-Vysoká úroveň</li> <li>Selhání:[errcode,]</li> </ul>

## 2.2.3.5.1 Příklad kódu

1	import Mrpc
2	#Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí ⊶objekt robota
3	robot = Mrpc.RPC('192.168.58.2')
4	di = robot.GetDI(0,0) # Získejte digitální vstup ovládacího pole
5	print(di)
5	print(di)

## 2.2.3.6 Získání digitálního vstupu nástroje

Prototyp	GetToolDI(id,block)
Popis	Získání digitálního vstupu nástroje
Parametr	<ul> <li>id:číslo IO,rozsah[0~1];</li> <li>blok: [0]- blokování, [1]- neblokující</li> </ul>
Návratová hodnota	<ul> <li>Úspěch:[0,di],di: 0-Nízká úroveň,1-Vysoká úroveň</li> <li>Selhání:[errcode,]</li> </ul>

### 2.2.3.6.1 Příklad kódu

1 (import Mrpc

2

3

4

# Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí ↔ objekt robota
robot = Mrpc.RPC('192.168.58.2')
tool\_di = robot.GetToolDI(1,0) # Získejte digitální vstup nástroje

₅ print(tool\_di)

## 2.2.3.7 Čekání na digitální vstup M om řídicí jednotky

Prototyp Popis	WaitDI(id,status,maxtime,opt) Čekání na digitální vstup M om řídicí jednotky
Parametr	<ul> <li>id:číslo IO,rozsah[0~15];</li> <li>stav:0-off,1-on;</li> <li>maxtime:Maximální čekací doba, jednotka[ms];</li> <li>opt:Po vypršení časového limitu strategie, 0-program se zastaví a vyzve k vypršení časového limitu, 1-ignorovat výzvu k vypršení časového limitu a pokračovat ve vykonávání programu, 2-pokračovat v čekání</li> </ul>
Návratová hodnota	<ul><li>Úspěch: [0]</li><li>Selhání:[errcode]</li></ul>

## 2.2.3.7.1 Příklad kódu

1 in	mport Mrpc					
2 #	Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí →objekt robota					
3 rc	obot = Mrpc.RPC('192.168.58.2')					
4 ro	obot.WaitDI(0,1,0,2) # Čekání na digitální vstup ovládacího panelu					
2.2.3.8	Čekání na	více	digitálních	vstupů M	om	řídicí skříňka
---------	-----------	------	-------------	----------	----	----------------
---------	-----------	------	-------------	----------	----	----------------

Prototyp	WaitMultiDI(mode,id,status,maxtime,opt)
Popis	Čekání na více digitálních vstupů M om řídicí skříňka
Parametr	<ul> <li>režim: [0]-Multiplex AND, [1]-Multiplex OR;</li> <li>id:číslo IO, bit0~bit7 odpovídá DI0~DI7, bit8~bit15 odpovídá CI0~CI7;</li> <li>status(uint16_t):bit0~bit7 odpovídá stavu DI0~DI7, bit8~bit15 odpovídá stavu DI0~DI7. reaguje na stavy stavových bitů CI0~CI7 [0]-off, [1]-on;</li> <li>maxtime:Maximální čekací doba, jednotka[ms];</li> <li>opt:Po vypršení časového limitu strategie, 0-program se zastaví a vyzve k vypršení časového limitu, 1-ignorovat výzvu k vypršení časového limitu a pokračovat ve vykonávání programu, 2-pokračovat v čekání</li> </ul>
Návratová hodnota	<ul><li>Úspěch: [0]</li><li>Selhání:[errcode]</li></ul>

## 2.2.3.8.1 Příklad kódu

import Mrpc 1

2

#Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí →objekt robota

- 3
- robot = Mrpc.RPC('192.168.58.2') robot.WaitMultiDI(1,3,3,10000,2) # Čekání na multiplexní digitální vstup řídicího boxu 4

## 2.2.3.9 Čekání na digitální vstup nástroje

Prototyp Popis	WaitToolDI(id,status,maxtime,opt) Čekání na koncový digitální vstup
Parametr	<ul> <li>id:číslo IO,rozsah[0~1];</li> <li>stav:0-off,1-on;</li> <li>maxtime:Maximální čekací doba, jednotka[ms];</li> <li>opt:po vypršení časového limitu strategie, 0-program se zastaví a vyzve k vypršení časového limitu, 1-ignorovat výzvu k vypršení časového limitu a pokračovat ve vykonávání programu, 2-pokračovat v čekání</li> </ul>
Návratová hodnota	<ul><li>Úspěch: [0]</li><li>Selhání:[errcode]</li></ul>

import Mrpc

#### 2.2.3.9.1 Příklad kódu

1 2

#Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí →objekt robota

robot = Mrpc.RPC('192.168.58.2') 3

robot.WaitToolDI(1,1,0,2) # Čekání na zadání čísla nástroje 4

## 2.2.3.10 Čekání na digitální vstup terminálu

Prototyp	GetAI(id,block)
Popis	Čekání na digitální vstup terminálu
Parametr	<ul> <li>id:číslo IO,rozsah[0~1];</li> </ul>
	<ul> <li>blok: [0]- blokování, [1]- neblokující</li> </ul>
Návratová hodnota	<ul> <li>Success:[0,value], value:Vstupní hodnota proudu nebo napětí v procentech, rozsah[0-100] odpovídá hodnotě proudu[0-20mA] nebo napětí[0-10V];</li> <li>Selhání:[errcode,]</li> </ul>

#### 2.2.3.10.1 Příklad kódu

1	import Mrpc
2	#Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí ↔objekt robota
3	robot = Mrpc.RPC('192.168.58.2')
4	ai = robot.GetAI(0,1) # Získejte analogový vstup řídicího boxu
	print(ci)

print(ai) 5

#### 2.2.3.11 Získání analogového vstupu nástroje

Prototyp Popis	GetToolAI(id,block) Získat terminálový analogový vstup
Parametr	<ul> <li>id:IO number,range[0];</li> <li>blok: [0]- blokování, [1]- neblokující</li> </ul>
Návratová hodnota	<ul> <li>Success:[0,value], value:Vstupní hodnota proudu nebo napětí v procentech, rozsah[0-100] odpovídá hodnotě proudu[0-20mA] nebo napětí[0-10V];</li> <li>Selhání:[errcode,]</li> </ul>

#### 2.2.3.11.1 Příklad kódu

1	import Mrpc
2	#Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení <mark>vrátí</mark> ↔objekt robota
3	robot = Mrpc.RPC('192.168.58.2')
4	tool_ai = robot.GetToolAI(0,1) # Získejte analogový vstup nástroje
5	print(tool_ai)

## 2.2.3.12 Čekání na simulaci řídicí jednotky vstup

Prototyp Popis	WaitAI(id,sign,value,maxtime,opt) Čekání na zadání simulace řídicí jednotky
Parametr	<ul> <li>id:číslo IO,rozsah[0~1];</li> <li>znaménko:0-Větší než,1-Méně než</li> <li>hodnota: Vstupní hodnota proudu nebo napětí v procentech, rozsah [0-100] odpovídá hodnotě proudu [0-20mA] nebo napětí [0-10V];</li> <li>maxtime:Maximální čekací doba, jednotka[ms];</li> <li>opt:Po vypršení časového limitu strategie, 0-program se zastaví a vyzve k vypršení časového limitu, 1-ignorovat výzvu k vypršení časového limitu a pokračovat ve vykonávání programu, 2-pokračovat v čekání</li> </ul>
Návratová hodnota	<ul><li>Úspěch: [0]</li><li>Selhání:[errcode]</li></ul>

#### 2.2.3.12.1 Příklad kódu

<sup>1</sup> (import Mrpc

- <sup>3</sup> robot = Mrpc.RPC('192.168.58.2')
- <sup>4</sup> robot.WaitAI(0,0,50,0,2) # *Vždy čeká na analogový vstup nástroje*

<sup>&</sup>lt;sup>2</sup> # Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí ↔ objekt robota

# 2.2.3.13 Čekání na analogový vstup nástroje

Prototyp	WaitToolAI(id,sign,value,maxtime,opt)
Popis	Čekání na koncový analogový vstup
Parametr	<ul> <li>id:IO number,range[0];</li> <li>znaménko:0-Větší než,1-Méně než</li> <li>hodnotu: Rozsah[0-100] odpovídá hodnotě proudu[0-20mA] nebo napětí[0-10V];</li> <li>maxtime:Maximální čekací doba, jednotka[ms];</li> <li>opt:Po vypršení časového limitu strategie, 0-program se zastaví a vyzve k vypršení časového limitu, 1-ignorovat výzvu k vypršení časového limitu a pokračovat ve vykonávání programu, 2-pokračovat v čekání</li> </ul>
Návratová hodnota	<ul><li>Úspěch: [0]</li><li>Selhání:[errcode]</li></ul>

### 2.2.3.13.1 Příklad kódu

1 2

3 4

import Mrpc	
#Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí _ ↔objekt robota	
robot = Mrpc.RPC('192.168.58.2')	
robot.WaitToolAI(0,0,50,0,2) # Vždy čeká na analogový vstup nástroje	

## 2.2.4 Společná nastavení

## 2.2.4.1 Nastavení globální rychlosti

Prototyp	SetSpeed(vel)
Popis	Nastavení globální rychlosti
Parametr	• vel:Procento rychlosti, rozsah[0~100]
Návratová hodnota	<ul><li>Úspěch: [0]</li><li>Selhání:[errcode]</li></ul>

## 2.2.4.1.1 Příklad kódu

1 (import Mrpc
2 # Je navázá

```
# Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí
→objekt robota
```

- <sup>3</sup> robot = Mrpc.RPC('192.168.58.2')
- <sup>4</sup> robot.SetSpeed(20) # Nastavte globální rychlost. Nastaví se ruční a automatický režim → nezávisle

#### 2.2.4.2 Nastavení hodnot systémových proměnných

Prototyp	SetSysVarValue(id,value)
Popis	Nastavení hodnot systémových proměnných
Parametr	<ul> <li>id:Číslo proměnné, rozsah[1~20];</li> <li>value:Hodnota proměnné</li> </ul>
Návratová hodnota	<ul><li>Úspěch: [0]</li><li>Selhání:[errcode]</li></ul>

## 2.2.4.2.1 Příklad kódu

1	import Mrpc
2	#Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí ↔objekt robota
3	robot = Mrpc.RPC('192.168.58.2')
4	for i in range(1,21):
5	robot.SetSysVarValue(i,i+0.5) # Nastavení hodnoty systémové proměnné
6	robot. WaitMs(1000)
7	for i in range(1,21):
8	sys_var = robot.GetSysVarValue(i) # Příklad Dotaz na hodnoty systémových proměnných
9	print(sys_var)

#### 2.2.4.3 Nastavení souřadnicového systému nástroje

Prototyp	SetToolCoord(id,t_coord,type,install)
Popis	Nastavení souřadnicového systému nástroje
Parametr	<ul> <li>id:Číslo souřadnicového systému, rozsah[0~14];</li> <li>t_coord:Poloha středového bodu nástroje vzhledem ke středu čelní příruby, jednotka[mm][°];</li> <li>typ:0-souřadnicový systém nástroje,1-souřadnicový systém senzoru;</li> <li>install:Instalační poloha,0-konec robota,1-vnější robot</li> </ul>
Návratová hodnota	<ul><li>Úspěch: [0]</li><li>Selhání:[errcode]</li></ul>

## 2.2.4.3.1 Příklad kódu

1 (import Mrpc

```
² # Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí
↔objekt robota
```

```
\rightarrow objekt robota
s robot = Mrpc.RPC('192.168.58.2')
```

- $\begin{array}{l} \text{robot} = \text{Mrpc.RPC}(192.168.58.27)\\ \text{t\_coord} = [1.0, 2.0, 3.0, 4.0, 5.0, 6.0] \end{array}$
- <sup>5</sup> robot.SetToolCoord(10,t\_coord,0,0) # Nastavení souřadnicového systému nástroje

## 2.2.4.4 Tabulka souřadnicových řad nástrojů

Prototyp Popis	SetToolList(id,t_coord , type,install) Tabulka souřadnicových řad nástrojů
Parametr	<ul> <li>id:Číslo souřadnicového systému, rozsah[0~14];</li> <li>t_coord:Poloha středového bodu nástroje vzhledem ke středu čelní příruby, jednotka[mm][°];</li> <li>typ:0-souřadnicový systém nástroje,1-souřadnicový systém senzoru;</li> <li>install:Instalační poloha 0-konec robota 1-vnější robot</li> </ul>
Návratová hodnota	<ul> <li>Úspěch: [0]</li> <li>Selhání:[errcode]</li> </ul>

## 2.2.4.4.1 Příklad kódu

1	import Mrpc
2	# Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí ↔objekt robota
3	robot = Mrpc.RPC('192.168.58.2')
4	$t\_coord = [1.0, 2.0, 3.0, 4.0, 5.0, 6.0]$
5	robot.SetToolList(10,t_coord,0,0) # Nastavení souřadnicového systému nástroje

## 2.2.4.5 Nastavení externího souřadnicového systému nástroje

Prototyp Popis	SetExToolCoord(id,etcp ,etool) Nastavení externího souřadnicového systému nástroje
Parametr	<ul> <li>id:Číslo souřadnicového systému, rozsah[0~14];</li> <li>etcp:Vnější souřadný systém nástroje, jednotka[mm][°];</li> <li>etool:Souřadnicový systém koncového nástroje, jednotka[mm][°];</li> </ul>
Návratová hodnota	<ul><li>Úspěch: [0]</li><li>Selhání:[errcode]</li></ul>

#### 2.2.4.5.1 Příklad kódu

import Mrpc

1

- #Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí →objekt robota 2
- robot = Mrpc.RPC('192.168.58.2') etcp 3 = [1.0,2.0,3.0,4.0,5.0,6.0]
- 4
- etool = [21.0,22.0,23.0,24.0,25.0,26.0] 5
- robot.SetExToolCoord(10,etcp,etool) 6

#### 2.2.4.6 Nastavení tabulky souřadnic externích nástrojů

Prototyp Popis Parametr	SetExToolList(id,etcp ,etool) Nastavení tabulky souřadnic externích nástrojů • id:Číslo souřadnicového systému, rozsah[0~14]; • etcp:Vnější souřadný systém nástroje, jednotka[mm][°]; • etool:Souřadnicový systém koncového nástroje, jednotka[mm][°];
Návratová hodnota	<ul> <li>Úspěch: [0]</li> <li>Selhání:[errcode]</li> </ul>

#### 2.2.4.6.1 Příklad kódu

1	import Mrpc
2	#Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí ⊶objekt robota
3	robot = Mrpc.RPC('192.168.58.2') etcp
4	= [1.0,2.0,3.0,4.0,5.0,6.0]
5	etool = [ 21.0,22.0,23.0,24.0,25.0,26.0]
6	robot.SetExToolList(10,etcp,etool)

#### 2.2.4.7 Nastavení souřadnicového systému obrobku

Prototyp	SetWObjCoord(id,w_coord)
Popis	Nastavení souřadnicového systému obrobku
Parametr	<ul> <li>id:Číslo souřadnicového systému, rozsah[0~14];</li> <li>w_coord:Relativní pozice souřadného systému, jednotka[mm][°];</li> </ul>
Návratová hodnota	<ul><li>Úspěch: [0]</li><li>Selhání:[errcode]</li></ul>

## 2.2.4.7.1 Příklad kódu

import Mrpc

1

2

```
#Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí 
→objekt robota
```

robot = Mrpc.RPC('192.168.58.2')

```
3
```

- w\_coord = [ 11.0,12.0,13.0,14.0,15.0,16.0] 4
- robot.SetWObjCoord(11,w\_coord) 5

## 2.2.4.8 Nastavení tabulky souřadnicových řad obrobků

Prototyp	SetWObjList(id,w_coord)
Popis	Nastavení tabulky souřadnicových řad obrobků
Parametr	<ul> <li>id:Číslo souřadnicového systému, rozsah[0~14];</li> <li>w_coord:Relativní pozice souřadného systému, jednotka[mm][°];</li> </ul>
Návratová hodnota	<ul><li>Úspěch: [0]</li><li>Selhání:[errcode]</li></ul>

## 2.2.4.8.1 Příklad kódu

import Mrpc 1

2

- #Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí →objekt robota
- robot = Mrpc.RPC('192.168.58.2') 3
- w\_coord = [ 11.0,12.0,13.0,14.0,15.0,16.0] robot.SetWObjList(11,w\_coord) 4
- 5

#### 2.2.4.9 Nastavení hmotnosti koncového zatížení

Prototyp	SetLoadWeight(hmotnost)
Popis	Nastavení hmotnosti koncového zatížení
Parametr	<ul> <li>hmotnost:jednotka[kg]</li> </ul>
Návratová hodnota	<ul><li>Úspěch: [0]</li><li>Selhání:[errcode]</li></ul>

## 2.2.4.9.1 Příklad kódu

	im
1	- <b>T</b> III

2

- port Mrpc # Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí ↔objekt robota
- 3
- robot = Mrpc.RPC('192.168.58.2') robot.SetLoadWeight(3.0) *# Nastavení hmotnosti nákladu* 4

#### 2.2.4.10 Nastavení způsobu instalace robota - pevná instalace

Prototyp	SetRobotInstallPos(metoda)
Popis	Nastavení způsobu instalace robota - pevná instalace
Parametr	• způsob:0-plochá instalace, 1-boční instalace, 2-závěsná instalace
Návratová hodnota	<ul><li>Úspěch: [0]</li><li>Selhání:[errcode]</li></ul>

## 2.2.4.10.1 Příklad kódu

1	import Mrpc
2	# Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí ⊶objekt robota
3	robot = Mrpc.RPC('192.168.58.2')
4	robot.SetRobotInstallPos(0) # Nastavení režimu instalace robota

#### 2.2.4.11 Nastavení úhlu instalace robota - M ee instalace

Prototyp	SetRobotInstallAngle(yangle,zangle)
Popis	Nastavení úhlu instalace robota - M ee instalace
Parametr	<ul> <li>yangle:Úhel oMoll</li> <li>zangle:Úhel natočení</li> </ul>
Návratová hodnota	<ul><li>Úspěch: [0]</li><li>Selhání:[errcode]</li></ul>

## 2.2.4.11.1 Příklad kódu

import Mrpc

1

2

- # Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí →objekt robota
- <sup>3</sup> robot = Mrpc.RPC('192.168.58.2')
- 4
   robot.SetRobotInstallAngle(0.0,0.0)
   #
   Nastavení instalačního úhlu robota

#### 2.2.4.12 Nastavení souřadnic centroidu koncového zatížení

Prototyp	SetLoadCoord(x,y,z)
Popis	Nastavení souřadnic centroidu koncového zatížení
Parametr	• x, y, z: Barycentrická souřadnice, jednotka[mm]
Návratová hodnota	<ul><li>Úspěch: [0]</li><li>Selhání:[errcode]</li></ul>

#### 2.2.4.12.1 Příklad kódu

1 (import Mrpc

2

```
# Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí __
↔objekt robota
```

- <sup>3</sup> robot = Mrpc.RPC('192.168.58.2')
- <sup>4</sup> robot.SetLoadCoord(3.0,4.0,5.0) # Nastavení souřadnic centroidu zátěže

#### 2.2.4.13 Čekání na zadaný čas

Prototyp	WaitMs(t_ms)
Popis	čekání na určitý čas
Parametr	• t_ms:unit[ms]
Návratová hodnota	<ul> <li>Úspěch: [0]</li> <li>Selhání:[errcode]</li> </ul>

#### 2.2.4.13.1 Příklad kódu

import Mrpc

1

2

- #Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí →objekt robota
- 3
- # Wait 1000ms 4

# 2.2.5 Nastavení zabezpečení

## 2.2.5.1 Nastavení úrovně kolize

Prototyp	SetAnticollision (mode,level,config)
Popis	Nastavení úrovně kolize
Parametr	<ul> <li>mode:0-level, 1-procentage;;</li> <li>level=[j1,j2,j3,j4,j5,j6]:práh kolize;</li> <li>config:0-neaktualizovat konfigurační soubor, 1-aktualizovat konfigurační soubor</li> </ul>
Návratová hodnota	<ul><li>Úspěch: [0]</li><li>Selhání:[errcode]</li></ul>

## 2.2.5.1.1 Příklad kódu

1	import Mrpc
2	#Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí ⊶objekt robota
3 4	robot = Mrpc.RPC('192.168.58.2') level = [1.0,2.0,3.0,4.0,5.0,6.0]
5 6	robot.SetAnticollision(0,level,1) # Nastavení úrovně kolize
7	robot.SetAnticollision(1,level,1) # Nastavení procenta kolize

#### 2.2.5.2 Nastavení strategie po kolizi

Prototyp	SetCollisionStrategy (strategie)
Popis	Nastavení strategie po kolizi
Parametr	<ul> <li>strategie:0-chybová pauza, 1-pokračovat v běhu</li> </ul>
Návratová hodnota	<ul><li>Úspěch: [0]</li><li>Selhání:[errcode]</li></ul>

## 2.2.5.2.1 Příklad kódu

import Mrpc

1

2

- #Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí ↔objekt robota
- <sup>3</sup> robot = Mrpc.RPC('192.168.58.2')
- <sup>4</sup> robot.SetCollisionStrategy(1) *# Nastavte strategii kolize po kolizi,1-Pokračovat v běhu*

#### 2.2.5.3 Nastavení kladného limitu

Prototyp	SetLimitPositive(p_limit)
Popis	Nastavení kladného limitu
Parametr	<ul> <li>p_limit=[j1,j2,j3,j4,j5,j6]:šest pozic kloubů</li> </ul>
Návratová hodnota	<ul><li>Úspěch: [0]</li><li>Selhání:[errcode]</li></ul>

#### 2.2.5.3.1 Příklad kódu

<sup>1</sup> (import Mrpc

2

- # Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí → objekt robota
- <sup>3</sup> robot = Mrpc.RPC('192.168.58.2')
- <sup>4</sup> p\_limit = [170.0,80.0,150.0,80.0,170.0,160.0]
- <sup>5</sup> robot.SetLimitPositive(p\_limit) # Nastavení kladného limitu

#### 2.2.5.4 Nastavení záporného limitu

Prototyp	SetLimitNegative(n_limit)
Popis	Nastavení záporného limitu
Parametr	<ul> <li>n_limit=[j1,j2,j3,j4,j5,j6]:šest pozic kloubů</li> </ul>
Návratová hodnota	<ul><li>Úspěch: [0]</li><li>Selhání:[errcode]</li></ul>

#### 2.2.5.4.1 Příklad kódu

import Mrpc

1

<sup>2</sup> # Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí

 $\leftrightarrow objekt \ robota$ robot = Mrpc.RPC('192.168.58.2')

robot = Mrpc.RPC('192.168.58.2')  $\lim_{n \to \infty} \lim_{n \to \infty}$ 

 $n\_limit = [-170.0, -260.0, -150.0, -260.0, -170.0, -160.0]$ 

```
s robot.SetLimitNegative(n_limit) # Nastavení záporného limitu
```

## 2.2.5.5 Vymazání chybového stavu

Prototyp	ResetAllError()
Popis	Stav chyby vymazán, lze vymazat pouze resetovatelné chyby.
Parametr	Nic
Návratová hodnota	<ul><li>Úspěch: [0]</li><li>Selhání:[errcode]</li></ul>

## 2.2.5.5.1 Příklad kódu

<sup>1</sup> (import Mrpc

2

# Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí ↔objekt robota

<sup>3</sup> robot = Mrpc.RPC('192.168.58.2')

<sup>4</sup> robot.ResetAllError() # Vymazání chybového stavu

## 2.2.5.6 Společný přepínač kompenzace m iction

Prototyp	MictionCompensationOnOff(state)
Popis	Společný přepínač kompenzace m iction
Parametr	• stav:0-off,1-on
Návratová hodnota	<ul> <li>Úspěch: [0]</li> <li>Selhání:[errcode]</li> </ul>

## 2.2.5.6.1 Příklad kódu

import Mrpc

1

2

- # Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí ↔objekt robota
- <sup>3</sup> robot = Mrpc.RPC('192.168.58.2')
- 4 robot.MictionCompensationOnOff(1) # Joint Miction compensation open

#### 2.2.5.7 Nastavení společného koeficientu kompenzace M iction formální instalace

Prototyp	SetMictionValue_level(coeff)
Popis	Nastavení společného koeficientu kompenzace M iction - formální instalace
Parametr	<ul> <li>coeff=[j1,j2,j3,j4,j5,j6]:šest společných kompenzačních koeficientů</li> </ul>
Návratová hodnota	<ul><li>Úspěch: [0]</li><li>Selhání:[errcode]</li></ul>

## 2.2.5.7.1 Příklad kódu

1	import Mrpc
2	# Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí ⇔ohiekt robota
3	robot = Mrpc.RPC(192.168.58.2)
4	<pre>robot.MictionCompensationOnOff(1) # Joint Miction compensation open</pre>
5	lcoeff = [0.9,0.9,0.9,0.9,0.9,0.9]
6	robot.SetMictionValue_level(lcoeff) # Nastavení společného koeficientu kompenzace M
	iction

#### 2.2.5.8 Nastavení koeficientu kompenzace kloubu M iction - Boční montáž

Prototyp	SetMictionValue_wall(coeff)
Popis	Nastavení koeficientu kompenzace kloubu M iction - Boční montáž
Parametr	<ul> <li>coeff=[j1,j2,j3,j4,j5,j6]:šest společných kompenzačních koeficientů</li> </ul>
Návratová hodnota	<ul><li>Úspěch: [0]</li><li>Selhání:[errcode]</li></ul>

## 2.2.5.8.1 Příklad kódu

1	import Mrpc
2	#Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí ⊶objekt robota
3	robot = Mrpc.RPC('192.168.58.2')
4	robot.MictionCompensationOnOff(1) # Joint Miction compensation open
5	wcoeff = [0,4,0,4,0,4,0,4,0,4]
6	robot.SetMictionValue_wall(wcoeff) # Nastavení společného koeficientu kompenzace M iction

## 2.2.5.9 Nastavení společného koeficientu kompenzace M - obráceně

Prototyp	SetMictionValue_ceiling(coeff)
Popis	Nastavení společného koeficientu kompenzace M - obráceně
Parametr	<ul> <li>coeff=[j1,j2,j3,j4,j5,j6]:šest společných kompenzačních koeficientů</li> </ul>
Návratová hodnota	<ul><li>Úspěch: [0]</li><li>Selhání:[errcode]</li></ul>

## 2.2.5.9.1 Příklad kódu

1	(import Mrpc
2	# Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí →objekt robota
3	robot = Mrpc.RPC('192.168.58.2')
4	robot.MictionCompensationOnOff(1) # Joint Miction compensation open
5	ccoeff = [0.6,0.6,0.6,0.6,0.6,0.6]
6	robot.SetMictionValue_ceiling(ccoeff) # Nastavení společného koeficientu kompenzace M
	iction

#### 2.2.5.10 Nastavení společného koeficientu kompenzace M iction-Mee instalace

Prototyp	SetMictionValue_Meedom(coeff)
Popis	Nastavení společného koeficientu kompenzace M iction-Mee instalace
Parametr	<ul> <li>coeff=[j1,j2,j3,j4,j5,j6]:šest společných kompenzačních koeficientů</li> </ul>
Návratová hodnota	<ul><li>Úspěch: [0]</li><li>Selhání:[errcode]</li></ul>

## 2.2.5.10.1 Příklad kódu

#### import Mrpc

1

- #Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí →objekt robota 2
- robot = Mrpc.RPC('192.168.58.2')
- 3 robot.MictionCompensationOnOff(1) # Joint Miction compensation open 4

```
5
   fcoeff = [0.5, 0.5, 0.5, 0.5, 0.5, 0.5]
```

robot.SetMictionValue\_Meedom(fcoeff) # Nastavení společného koeficientu kompenzace M 6 iction

# 2.2.6 Stavový dotaz

## 2.2.6.1 Získání úhlu instalace robota

Prototyp	GetRobotInstallAngle()
Popis	Získání úhlu instalace robota
Parametr	Nic
Návratová hodnota	<ul> <li>Úspěch:[0,yangle,zangle],yangle-úhel oMoll,zangle-úhel rotace</li> <li>Selhání:[errcode,]</li> </ul>

## 2.2.6.1.1 Příklad kódu

1	import Mrpc
2	#Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení <mark>vrátí _</mark> ⊶objekt robota
3	robot = Mrpc.RPC('192.168.58.2')
4	ret = robot.GetRobotInstallAngle() # Získejte úhel instalace robota
5	print(ret)

#### 2.2.6.2 Získání hodnot systémových proměnných

Prototyp	GetSysVarValue(id)
Popis	Získání hodnot systémových proměnných
Parametr	<ul> <li>id:Číslo systémové proměnné, rozsah[1~20]</li> </ul>
Návratová hodnota	<ul><li>Úspěch:[0,var_value]</li><li>Selhání:[errcode,]</li></ul>

### 2.2.6.2.1 Příklad kódu

```
import Mrpc
1
  # Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí
2
   ·→objekt robota
  robot = Mrpc.RPC('192.168.58.2')
3
  for i in range(1,21):
4
      robot.SetSysVarValue(i,i+0.5) # Nastavení hodnoty systémové proměnné
5
  robot.WaitMs(1000)
6
7
  for i in range(1,21):
      sys_var = robot.GetSysVarValue(i) # Dotaz na hodnoty systémových proměnných
8
      print(sys_var)
9
```

#### 2.2.6.3 Zjištění aktuální polohy kloubu (úhel)

Prototyp	GetActualJointPosDegree(flag)
Popis	Zjištění aktuální polohy (úhlu) kloubu)
Parametr	<ul> <li>příznak:0-blokování, 1-neblokování</li> </ul>
Návratová hodnota	<ul><li>Úspěch:[0,joint_pos],joint_pos=[j1,j2,j3,j4,j5,j6]</li><li>Selhání:[errcode,]</li></ul>

## 2.2.6.3.1 Příklad kódu

1	(import Mrpc
2	#Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí ↔objekt robota
3	robot = Mrpc.RPC('192.168.58.2')
4	ret = robot.GetActualJointPosDegree(0)  # Získejte aktuální polohu kloubu robota
5	print(ret)

## 2.2.6.4 Získání aktuální polohy kloubu (radián)

Prototyp	GetActualJointPosRadian(flag)
Popis	Získání aktuální polohy kloubu (radián)
Parametr	<ul> <li>příznak:0-blokování, 1-neblokování</li> </ul>
Návratová hodnota	<ul> <li>Úspěch:[0,joint_pos],joint_pos=[j1,j2,j3,j4,j5,j6]</li> <li>Selhání:[errcode,]</li> </ul>

#### 2.2.6.4.1 Příklad kódu

import Mrpc
# Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí
c→objekt robota
robot = Mrpc.RPC('192.168.58.2')
ret = robot.GetActualJointPosRadian(0) # Získejte aktuální polohu kloubu robota
print(ret)

#### 2.2.6.5 Získání aktuální polohy nástroje

Prototyp	GetActualTCPPose(flag)
Popis	Získání aktuální polohy nástroje
Parametr	<ul> <li>příznak:0-blokování, 1-neblokování</li> </ul>
Návratová hodnota	<ul> <li>Úspěch:[0,tcp_pose],tcp_pose=[x,y,z,rx,ry,rz]</li> <li>Selhání:[errcode,]</li> </ul>

#### 2.2.6.5.1 Příklad kódu

1	import Mrpc
2	#Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí →objekt robota
3	robot = Mrpc.RPC('192.168.58.2')
4	ret = robot.GetActualTCPPose(0)  # Získejte aktuální pozici robota jako nástroje
5	print(ret)

#### 2.2.6.6 Získání aktuálního čísla souřadnicového systému nástroje

Prototyp	GetActualTCPNum(flag)
Popis	Získání aktuálního čísla souřadnicového systému nástroje
Parametr	<ul> <li>příznak:0-blokování, 1-neblokování</li> </ul>
Návratová hodnota	<ul> <li>Úspěch:[0,tool_id]</li> <li>Selhání:[errcode,]</li> </ul>

### 2.2.6.6.1 Příklad kódu

1	import Mrpc
2	#Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí ∟ ↔objekt robota
3	robot = Mrpc.RPC('192.168.58.2')
4	ret = robot.GetActualTCPNum(0) # Získejte aktuální číslo souřadného systému nástroje
5	print(ret)

## 2.2.6.7 Získání aktuálního čísla souřadnicového systému obrobku

Prototyp Popis	GetActualWObjNum(flag) Získání aktuálního čísla souřadnicového svstému obrobku
Parametr	<ul> <li>příznak:0-blokování, 1-neblokování</li> </ul>
Návratová hodnota	<ul><li>Úspěch:[0,wobj_id]</li><li>Selhání:[errcode,]</li></ul>

#### 2.2.6.7.1 Příklad kódu

import Mrpc
# Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí
c→objekt robota
robot = Mrpc.RPC('192.168.58.2')
ret = robot.GetActualWObjNum(0) # Získejte aktuální číslo souřadnicového systému obrobku
print(ret)

## 2.2.6.8 Zjištění aktuální polohy koncové příruby

Prototyp	GetActualToolFlangePose(flag)
Popis	Zjištění aktuální polohy koncové příruby
Parametr	<ul> <li>příznak:0-blokování, 1-neblokování</li> </ul>
Návratová hodnota	<ul> <li>Úspěch:[0,flange_pose],flange_pose=[x,y,z,rx,ry,rz]</li> <li>Selhání:[errcode,]</li> </ul>

#### 2.2.6.8.1 Příklad kódu

import Mrpc
# Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí
c→objekt robota
robot = Mrpc.RPC('192.168.58.2')
ret = robot.GetActualToolFlangePose(0) # Získejte aktuální pozici koncové příruby
print(ret)

## 2.2.6.9 Řešení inverzní kinematiky

Prototyp	GetInverseKin(type,desc_pos,config)
Popis	Inverzní kinematika, kartézská pozice pro řešení polohy kloubu
Parametr	<ul> <li>typ:0-absolutní pozice (základní souřadný systém), 1-relativní pozice (základní souřadný systém), 2-relativní pozice (souřadný systém nástroje)</li> <li>desc_pose:[x,y,z,rx,ry,rz],pozice nástroje,jednotka[mm][°]</li> <li>config:Konfigurace kloubu, [-1]-vztahuje se k aktuální poloze kloubu pro řešení, [0-7]-řeší se na základě konfigurace kloubu</li> </ul>
Návratová hodnota	<ul> <li>Úspěch:[0,joint_pos],joint_pos=[j1,j2,j3,j4,j5,j6]</li> <li>Selhání:[errcode,]</li> </ul>

#### 2.2.6.9.1 Příklad kódu

1	(import Mrpc
2	# Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí ↔objekt robota
3 4	robot = Mrpc.RPC('192.168.58.2') P1=[75.414,568.526,338.135,-178.348,-0.930,52.611]
5 6	ret = robot.GetInverseKin(0,P1,-1) print(ret)

Prototyp	GetInverseKinRef(type,desc_pos,joint_pos_ref)
Popis	Inverzní kinematika řeší inverzní kinematiku, pozice nástroje řeší polohu kloubu a odkazuje na inverzní kinematiku. na zadanou polohu kloubu pro řešení
Parametr	<ul> <li>typ:0-absolutní pozice (základní souřadný systém), 1-relativní pozice (základní souřadný systém), 2-relativní pozice (souřadný systém nástroje)</li> <li>desc_pos:[x,y,z,rx,ry,rz]pozice nástroje,jednotka[mm][°]</li> </ul>
	<ul> <li>joint_pos_ref:[j1,j2,j3,j4,j5,j6], referenční poloha kloubu, jednotka[°]</li> </ul>
Návratová hodnota	<ul> <li>Úspěch:[0,joint_pos],joint_pos=[j1,j2,j3,j4,j5,j6]</li> <li>Selhání:[errcode,]</li> </ul>

## 2.2.6.10 Řešení inverzní kinematiky - Zadání referenční polohy

#### 2.2.6.10.1 Příklad kódu

1	import Mrpc
2	# Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí ↔objekt robota
3	robot = Mrpc.RPC('192.168.58.2')
4	P1=[75.414,568.526,338.135,-178.348,-0.930,52.611]
5	J1=[95.442,-101.149,-98.699,-68.347,90.580,-47.174]
6	ret = robot.GetInverseKinRef(0,P1,J1)
7	print(ret)

## 2.2.6.11 Řešení inverzní kinematiky - zda existuje řešení

Prototyp	GetInverseKinHasSolution(type,desc_pos,joint_pos_ref)
Popis	Inverzní kinematika, řešení polohy nástroje, zda je řešena poloha kloubu
Parametr	<ul> <li>typ:0-Absolutní poloha (základní souřadnicový systém), 1-Relativní poloha (základní souřadnicový systém), 2-Relativní poloha (souřadnicový systém nástroje)</li> <li>desc_pos:[x,y,z,rx,ry,rz]pozice nástroje, jednotka[mm][°]</li> <li>joint_pos_ref:[j1,j2,j3,j4,j5,j6],referenční poloha kloubu, jednotka[°]</li> </ul>
Návratová hodnota	<ul> <li>Úspěch:[0,result], "True"-s řešením, "False"-bez řešení</li> <li>Selhání:[errcode,]</li> </ul>

#### 2.2.6.11.1 Příklad kódu

import Mrpc

```
1
   # Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí
2
   \leftrightarrow objekt robota
   robot = Mrpc.RPC('192.168.58.2')
3
   P1=[75.414,568.526,338.135,-178.348,-0.930,52.611]
4
   J1=[95.442,-101.149,-98.699,-68.347,90.580,-47.174]
5
   ret = robot.GetInverseKinHasSolution(0,P1,J1)
6
   print(ret)
7
```

## 2.2.6.12 Řešení dopředné kinematiky

Prototyp	GetForwardKin(joint_pos)
Popis	dopředná kinematika, řešení polohy kloubů, póza nástroje
Parametr	<ul> <li>joint_pos:[j1,j2,j3,j4,j5,j6]:kloub Poloha kloubu,jednotka[°]</li> </ul>
Návratová hodnota	<ul> <li>Úspěch:[0,desc_pos],desc_pos=[x,y,z,rx,ry,rz]:pozice nástroje,jednotka[mm][°]</li> <li>Selhání:[errcode,]</li> </ul>

### 2.2.6.12.1 Příklad kódu

1

2

3

4

5

6

```
import Mrpc
# Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí
↔objekt robota
robot = Mrpc.RPC('192.168.58.2')
J1=[95.442,-101.149,-98.699,-68.347,90.580,-47.174]
ret = robot.GetForwardKin(J1)
print(ret)
```

#### 2.2.6.13 Zjištění aktuálního kloubního momentu

Prototyp	GetJointTorques(flag)
Popis	Zjištění aktuálního kloubního momentu
<ul> <li>Parametr</li> <li>příznak:0-blokování, 1-neblokování</li> </ul>	
Návratová hodnota	<ul> <li>Úspěch:[0,torques],torques=[j1,j2,j3,j4,j5,j6]</li> <li>Selhání:[errcode,]</li> </ul>

## 2.2.6.13.1 Příklad kódu

1	(import Mrpc			
2	#Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí ↔objekt robota			
3	robot = Mrpc.RPC('192.168.58.2')			
4	ret = robot.GetJointTorques(0) # Získejte aktuální kloubní moment			
5	print(ret)			

#### 2.2.6.14 Zjištění hmotnosti aktuálního nákladu

Prototyp	GetTargetPayload(flag)
Popis	Zjištění hmotnosti aktuálního nákladu
Parametr	<ul> <li>příznak:0-blokování, 1-neblokování</li> </ul>
Návratová hodnota	<ul><li>Úspěch:[0,hmotnost],jednotka[kg]</li><li>Selhání:[errcode,]</li></ul>

## 2.2.6.14.1 Příklad kódu

1	import Mrpc
2	#Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí _ ↔objekt robota
3	robot = Mrpc.RPC('192.168.58.2')
4	ret = robot.GetTargetPayload(0) # Získejte hmotnost aktuálního nákladu
5	print(ret)

## 2.2.6.15 Získání centroidu aktuálního zatížení

Prototyp	GetTargetPayloadCog(flag)
Popis	Získání centroidu aktuálního zatížení
Parametr	<ul> <li>příznak:0-blokování, 1-neblokování</li> </ul>
Návratová hodnota	<ul> <li>Úspěch:[0,cog], cog=[x,y,z]:barycentrická souřadnice,jednotka[mm]</li> <li>Selhání:[errcode,]</li> </ul>

#### 2.2.6.15.1 Příklad kódu

1 import Mrpc

2

```
# Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí

↔objekt robota
```

<sup>3</sup> robot = Mrpc.RPC('192.168.58.2')

```
<sup>4</sup> ret = robot.GetTargetPayloadCog(0) # Získejte centroid aktuální zátěže
```

₅ print(ret)

## 2.2.6.16 Získání aktuálního souřadnicového systému nástroje

Prototyp	GetTCPOffset(flag)			
Popis	Získání aktuálního souřadnicového systému nástroje			
Parametr	<ul> <li>příznak:0-blokování, 1-neblokování</li> </ul>			
Návratová hodnota	<ul> <li>Success:[0,tcp_offset], tcp_offset=[x,y,z,rx,ry,rz]:,unit[mm][°]</li> <li>Selhání:[errcode,]</li> </ul>			

## 2.2.6.16.1 Příklad kódu

1	import Mrpc
2	#Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí ⊶objekt robota
3	robot = Mrpc.RPC('192.168.58.2')
4	ret = robot.GetTCPOffset(0) # Získejte aktuální souřadnicový systém nástroje
5	print(ret)

## 2.2.6.17 Získání aktuálního souřadnicového systému obrobku

Prototyp	GetWObjOffset(flag)			
Popis	Získání aktuálního souřadnicového systému obrobku			
<ul> <li>Parametr</li> <li>příznak:0-blokování, 1-neblokování</li> </ul>				
Návratová hodnota	<ul> <li>Success:[0,wobj_offset], wobj_offset=[x,y,z,rx,ry,rz]:relativní pozice,jednotka[mm][°]</li> <li>Selhání:[errcode,]</li> </ul>			

#### 2.2.6.17.1 Příklad kódu

1	import Mrpc
2	#Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení <mark>vrátí</mark> ⊶objekt robota
3	robot = Mrpc.RPC('192.168.58.2')
4	ret = robot.GetWObjOffset(0) # Získejte aktuální souřadnicový systém obrobku
5	print(ret)

## 2.2.6.18 Získat měkký mezní úhel kloubu

Drototyp	Cotloints of Limit Dog(flag)	
Рююур	GetJointSoftLinntDeg(nag)	
Popis	Získat měkký mezní úhel kloubu	
Parametr     - flag:0-blokování, 1-neblokování       Vrátit value     Graves 10. i4. is i1. se in i2. se in i2. se i1. se i1		
Vrátit value	<ul> <li>Success:[0, j1min,j1max,j2min,j2max,j3min,j3max,j4min,j4max,j5min,j5max,j6m :osa 1 až osa 6 záporná mez a kladná mez, jednotka[mm]</li> <li>Selhání:[errcode,]</li> </ul>	in,j6max]

#### 2.2.6.18.1 Příklad kódu

1 (import	Mrpc
-----------	------

2

11 T		· /	×11.1.1.1	1 (	$\mathbf{T}_{\mathbf{T}}$ $\mathbf{v}\mathbf{v}$ $\mathbf{v}$ $\mathbf{v}$	
<i>∓Je n</i>	avazano	spojeni s	<i>гіаісі јеапо</i> ікои	robota.	Uspesne spojeni Vrati	
1	ist wol	1 5	5		1 1 5	
$\rightarrow OC$	ekt rode	<i>71a</i>				

- 3
- robot = Mrpc.RPC('192.168.58.2')
  ret = robot.GetJointSoftLimitDeg(0) # získáme měkký mezní úhel kloubu
  print(ret) 4
- 5

## 2.2.6.19 Získat systémový čas

Prototyp	GetSystemClock()
Popis	Získat systémový čas
Parametr	Nic
Návratová hodnota	<ul><li>Úspěch:[0,t_ms]:jednotka[ms]</li><li>Selhání:[errcode,]</li></ul>

#### 2.2.6.19.1 Příklad kódu

<sup>1</sup> import Mrpc

2

Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí∟ →objekt robota	

- <sup>3</sup> robot = Mrpc.RPC('192.168.58.2')
- 4 ret = robot.GetSystemClock() # Získejte systémový čas řídicí jednotky
- 5 print(ret)

#### 2.2.6.20 Zjištění aktuální konfigurace kloubů robota

Prototyp	GetRobotCurJointsConfig()
Popis	Zjištění aktuální konfigurace kloubů robota
Parametr	Nic
Návratová hodnota	<ul><li>Úspěch:[0,config]:range[0~7]</li><li>Selhání:[errcode,]</li></ul>

#### 2.2.6.20.1 Příklad kódu

- 1 import Mrpc
- <sup>2</sup> *# Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení* vrátí → *objekt robota*
- <sup>3</sup> robot = Mrpc.RPC('192.168.58.2')
- 4 ret = robot.GetRobotCurJointsConfig() # Ziskejte aktuálni konfiguraci kloubů\_
- *∽robot*
- 5 print(ret)

#### 2.2.6.21 Získat výchozí rychlost

Prototyp	GetDefaultTransVel()
Popis	Získat výchozí rychlost
Parametr	Nic
Návratová hodnota	<ul> <li>Úspěch:[0,vel]:jednotka[mm/s]</li> <li>Selhání:[errcode,]</li> </ul>

#### 2.2.6.21.1 Příklad kódu

1	import Mrpc
2	#Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí ↔objekt robota
3	robot = Mrpc.RPC('192.168.58.2')
4	ret = robot.GetDefaultTransVel() # Zjistí aktuální rychlost robota
5	print(ret)
5	princiec)

#### 2.2.6.22 Zkontrolujte, zda je pohyb robota dokončen

Prototyp	GetRobotMotionDone()
Popis	Zkontrolujte, zda je pohyb robota dokončen
Parametr	Nic
Návratová hodnota	<ul><li>Úspěch:[0,stav],stav:0-nedokončeno,1-nedokončeno</li><li>Selhání:[errcode,]</li></ul>

#### 2.2.6.22.1 Příklad kódu

<sup>1</sup> (import Mrpc

```
<sup>2</sup> # Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí 
↔ objekt robota
```

```
<sup>3</sup> robot = Mrpc.RPC('192.168.58.2')
```

```
<sup>4</sup> ret = robot.GetRobotMotionDone() #Dotaz na stav dokončení pohybu robota
```

s if ret[0] == 0:

- 6 print(ret[1])
- , jinak:

8

print("errcode je: ", ret[0])

# 2.2.7 Opakování trajektorie

# 2.2.7.1 Nastavení parametrů záznamu trajektorie

Prototyp Popis	SetTPDParam(type,name,period_ms,di_choose,do_choose) Nastavení parametrů záznamu trajektorie	
Parametr	<ul> <li>typ:Datový typ, 1-spojová pozice;</li> <li>název:Název skladby;</li> <li>period_ms:Perioda vzorkování, pevná hodnota, 2 ms nebo 4 ms nebo 8 ms;</li> <li>di_choose:výběr DI, bit0~bit7 odpovídá řídicím boxům DI0~DI7, bit8~bit9 odpovídá terminálu DI0~DI1, 0-nezvoleno, 1-zvoleno</li> <li>do_choose:výběr DO, bit0~bit7 odpovídá řídicím boxům DO0~DO7, bit8~bit9 odpovídá terminálu DO0~DO1, 0-není vybráno, 1-je vybráno</li> </ul>	
Návratová hodnota	<ul><li>Úspěch: [0]</li><li>Selhání:[errcode]</li></ul>	

## 2.2.7.1.1 Příklad kódu

1 (i	import Mrpc
2 \$	#Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí∟ ↔objekt robota
3 r	robot = Mrpc.RPC('192.168.58.2')
4 t	type = 1 # Datový typ, pozice 1 kloubu
5 ľ	name = 'tpd2023' # <i>Název stopy</i>
6 C	období = 4    # Perioda vzorkování, pevná hodnota, 2ms nebo 4ms nebo 8ms
7 C	di_choose = 0 # konfigurace di vstupu
8 C	do_choose = 0 # proveďte konfiguraci výstupu
9 r	robot.SetTPDParam(type, name, period, di_choose, do_choose) #Konfigurace parametru TPD

## 2.2.7.2 Spuštění záznamu trajektorie

Prototyp	SetTPDStart(type,name,period_ms,di_choose,do_choose)
Popis	Spuštění záznamu trajektorie
Parametr	<ul> <li>typ:Datový typ, 1-spojová pozice;</li> <li>název:Název skladby;</li> <li>period_ms:Perioda vzorkování, pevná hodnota, 2 ms nebo 4 ms nebo 8 ms;</li> <li>di_choose:výběr DI, bit0~bit7 odpovídá řídicím boxům DI0~DI7, bit8~bit9 odpovídá terminálu DI0~DI1, 0-nezvoleno, 1-zvoleno</li> <li>do_choose:výběr DO, bit0~bit7 odpovídá řídicím boxům DO0~DO7, bit8~bit9 odpovídá terminálu DO0~DO1, 0-není vybráno, 1-je vybráno</li> </ul>
Návratová hodnota	<ul> <li>Úspěch: [0]</li> <li>Selhání:[errcode]</li> </ul>

## 2.2.7.3 Zastavení záznamu trajektorie

Prototyp	SetWebTPDStop()
Popis	Zastavení záznamu trajektorie
Parametr	Nic
Návratová hodnota	<ul><li>Úspěch: [0]</li><li>Selhání:[errcode]</li></ul>

# 2.2.7.3.1 Příklad kódu

1	import Mrpc
2	import time
3	# Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí ↔objekt robota
4	robot = Mrpc.RPC('192.168.58.2')
5	type = 1 # Datový typ, pozice 1 kloubu
6	name = 'tpd2023' # Název stopy
7	období = 4 # Perioda vzorkování, pevná hodnota, 2ms nebo 4ms nebo 8ms
8	di_choose = 0 # konfigurace di vstupu
9	do_choose = 0 # proveďte konfiguraci výstupu
10	robot.SetTPDParam(type, name, period, di_choose, do_choose) #Konfigurace parametru TPD
11	robot.Mode(1) # Robot přejde do manuálního režimu
12	time.sleep(1)
13	robot.DragTeachSwitch(1) #Robot přejde do režimu výuky přetahováním
14	robot.SetTPDStart(type, name, period, di_choose, do_choose) # <i>Spust'te</i> nahrávání _ <i>→výukový kurz</i>
15	time.sleep(30)
16	robot.SetWebTPDStop() # Zastavení nahrávání výukových stop
17	robot.DragTeachSwitch(0) #Robot přejde do režimu výuky bez tahání

## 2.2.7.4 Odstranění záznamu trajektorie

Prototyp	SetTPDDelete(name)	
Popis	Odstranění záznamu trajektorie	
Parametr	• název:Název skladby	
Návratová hodnota	<ul><li>Úspěch: [0]</li><li>Selhání:[errcode]</li></ul>	

#### 2.2.7.4.1 Příklad kódu

import Mrpc

1

2

- #Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí ↔objekt robota
- <sup>3</sup> robot = Mrpc.RPC('192.168.58.2')
- 4 robot.SetTPDDelete('tpd2023') # Odstranění stopy TPD

#### 2.2.7.5 Přednačítání trajektorie

Prototyp	LoadTPD(name)
Popis	Přednačítání trajektorie
Parametr	• název:Název skladby
Návratová hodnota	<ul><li>Úspěch: [0]</li><li>Selhání:[errcode]</li></ul>

#### 2.2.7.6 Reprodukce trajektorie

Prototyp Popis	MoveTPD(name,blend,ovl) Reprodukce trajektorie
Parametr	<ul> <li>název:Název skladby</li> <li>směs:Je hladká, 0-není hladká, 1-hladká</li> <li>ovl:Faktor škálování rychlosti, rozsah[0~100]</li> </ul>
Návratová hodnota	<ul><li>Úspěch: [0]</li><li>Selhání:[errcode]</li></ul>

#### 2.2.7.6.1 Příklad kódu

```
import Mrpc
1
   # Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí
2
   ⊶objekt robota
   robot = Mrpc.RPC('192.168.58.2')
3
   P1=[-378.9,-340.3,107.2,179.4,-1.3,125.0]
4
   n\acute{a}zev = "tpd2023
                        #Název dráhy
5
                #Je hladký, 0-není hladký, 1-hladký
   sm\check{e}s = 1
6
   ovl = 100,0
                   #Škálování rychlosti
7
   robot.LoadTPD(name) #Přednačítání trajektorie
8
   robot.MoveCart(P1,1,0,100.0,100.0,100.0,-1.0,-1)
                                                                #Přejděme k výchozímu bodu
9
   robot.MoveTPD(name, blend, ovl)
                                        #Reprodukce trajektorie
10
```

# 2.2.8 Použití programu WebAPP

#### 2.2.8.1 Nastavení a automatické načtení výchozího operačního programu

Prototyp Popis	LoadDefaultProgConfig(flag,program_name) Nastavení a automatické načtení výchozího operačního programu
Parametr	<ul> <li>příznak:1-automatické načtení výchozího programu při spuštění, 0- neautomatické načtení výchozího programu</li> <li>program_name:Název a cesta k programu domácího úkolu, např. "/Maser/movej.lua", kde "/Maser/" je pevná cesta.</li> </ul>
Návratová hodnota	<ul><li>Úspěch: [0]</li><li>Selhání:[errcode]</li></ul>

#### 2.2.8.1.1 Příklad kódu

#### <sup>1</sup> (import Mrpc

2

# Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí \_\_ ↔objekt robota

- <sup>3</sup> robot = Mrpc.RPC('192.168.58.2')
- robot.LoadDefaultProgConfig(1, "/Muser/splineptp.lua") # Nastavte výchozí program úlohy\_
   → aby se automaticky načetl při spuštění systému

#### 2.2.8.2 Načtení zadaného programu úlohy

Prototyp	ProgramLoad(n <b>á</b> zev_programu)	
Popis	Načtení zadaného programu úlohy	
Parametr	<ul> <li>program_name:Název a cesta k programu domácího úkolu, "/Maser/movej.lua", kde "/Maser/" je pevná cesta.</li> </ul>	jako např.
Návratová hodnota	<ul><li>Úspěch: [0]</li><li>Selhání:[errcode]</li></ul>	

#### 2.2.8.2.1 Příklad kódu

1 import Mrpc

2

# Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí ↔ objekt robota

- <sup>3</sup> robot = Mrpc.RPC('192.168.58.2')
- 4 #Robot webapp program používá rozhraní
- <sup>5</sup> robot.Mode(0) #*Robot přejde do automatického provozního režimu*

(pokračování na další straně)

6

(pokračování na předchozí straně)

robot.ProgramLoad('/Muser/testPTP.lua') #Chcete-li načíst program robota, který se má ↔ program testPTP.lua musí být nejprve napsán ᠳPustiebotiskněte tlačítko\_ aplikaci

#### 2.2.8.3 Získání čísla řádku provádění aktuálního programu úlohy robota.

Prototyp	GetCurrentLine()
Popis	Získání čísla řádku provádění aktuálního programu úlohy robota.
Parametr	Nic
Návratová hodnota	<ul><li>Úspěch:[0,line_num]</li><li>Selhání:[errcode]</li></ul>

#### 2.2.8.4 Spustit aktuálně načtený program úlohy

Prototyp	ProgramRun()
Popis	Spustit aktuálně načtený program úlohy
Parametr	Nic
Návratová hodnota	<ul> <li>Úspěch: [0]</li> <li>Selhání:[errcode]</li> </ul>

#### 2.2.8.5 Pozastavení aktuálně spuštěného pracovního programu

Prototyp	DrogramDauso
гююцур	riogrammause()
Popis	Pozastavení aktuálně spuštěného pracovního programu
Parametr	Nic
Návratová hodnota	<ul><li>Úspěch: [0]</li><li>Selhání:[errcode]</li></ul>

#### 2.2.8.6 Obnovení aktuálně pozastaveného pracovního programu

Prototyp	ProgramResume()
Popis	Obnovení aktuálně pozastaveného pracovního programu
Parametr	Nic
Návratová hodnota	<ul> <li>Úspěch: [0]</li> <li>Selhání:[errcode]</li> </ul>

## 2.2.8.7 Ukončení aktuálně spuštěného programu úlohy

Prototyp	Program Stop()
Ponis	Ilkončení aktuálně snuštěného programu úlohy
Parametr	Nic
Návratová bodnota	
Naviatova nocilota	• Úspěch: [0]
	• Selhání:[errcode]

# 2.2.8.8 Zjištění stavu provádění programů úloh oMobot

Prototyp	GetProgramState()
Popis	Zjištění stavu provádění programů úloh oMobot
Parametr	Nic
Návratová hodnota	<ul> <li>Success:[0,state],state:1-program zastaven nebo žádný program neběží, 2- program běží, 3-program pozastaven</li> <li>Selhání:[errcode]</li> </ul>

## 2.2.8.9 Zjištění názvu načteného programu úlohy

Prototyp	GetLoadedProgram()
Popis	Zjištění názvu načteného programu úlohy
Parametr	Nic
Návratová hodnota	<ul><li>Úspěch:[0,název_programu]</li><li>Selhání:[errcode]</li></ul>

## 2.2.8.9.1 Příklad kódu

1	import Mrpc
2	import time
3	import
4	_thread
5	def print_program_state(name,rb):
6	whips(a)e = robot.GetProgramState() #Dotaz na stav spuštěného programu,1-
	↔zastaveno nebo nic program běží, 2-program běží, PppB54Um-pozastaveno
7	linenum = robot.GetCurrentLine() #Dotaz na číslo řádku aktuální úlohy_
	(→program
8	name = robot.GetLoadedProgram() #Dotaz na název načteného pracovního
9	print("stav programu robota je:",pstate[%])##int("číslo
10	řádku programu robota je:",linenum[1]) print("název
1	programu robota j e : ",name[1])
12	time.sleep(1)

(pokračování na další straně)

(pokračování na předchozí straně)

13	#Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení <mark>vrátí</mark> ⊶objekt robota
14 15 16	robot = Mrpc.RPC('192.168.58.2') #Robot webapp program používá rozhraní robot.Mode(0) #Robot vstoupil do automatického provozního režimu
17	robot.ProgramLoad('/Muser/testPTP.lua') #Pro načtení programu robota, který se má spustit,
18	se použije příkaz ↔program <sub>testerp</sub> .lua musí být nejprve napsán ve webové aplikaci
19	robot.ProgramRun() #Výkonný program robota
20	timfead start new thread print program state. ("print_state", robot))
21	robot.ProgramPause() #Pozastavení probíhajícího programu
22	time.sleep(5) robota
23	robot.ProgramResume() #Obnovení pozastaveného programu
24	time.sleep(5) robota
25	robot.ProgramStop()
26	time.sleep(2) #Zastavení probíhajícího programu
	robota

# 2.2.9 Periferní

## 2.2.9.1 Získání konfigurace chapadla

Prototyp	GetGripperConfig()
Popis	Získání konfigurace chapadla
Parametr	Nic
Návratová hodnota	<ul> <li>Úspěch:[0, společnost,zařízení,softverze,sběrnice],společnost:</li> <li>Selhání:[errcode]</li> </ul>

## 2.2.9.2 Aktivace chapadla

Prototyp	ActGripper(index,action)
Popis	Aktivace chapadla
Parametr	<ul> <li>index:Číslo drápu;</li> <li>akce: 0-resetovat, 1-aktivovat</li> </ul>
Návratová hodnota	<ul><li>Úspěch: [0]</li><li>Selhání:[errcode]</li></ul>

# 2.2.9.3 Řídicí chapadlo

Prototyp Popis	MoveGripper(index,pos,speed,force,maxtime,block) Řídicí chapadlo
Parametr	<ul> <li>index:Číslo drápu;</li> <li>pos:Procento polohy, rozsah[0~100];</li> <li>speed:Procento rychlosti, rozsah[0~100];</li> <li>force:Moment v procentech, rozsah[0~100];</li> <li>maxtime:Maximální čekací doba, rozsah[0~30000],jednotka[ms];</li> <li>blokování:0-blokování, 1-neblokování</li> </ul>
Návratová hodnota	<ul> <li>Úspěch: [0]</li> <li>Selhání:[errcode]</li> </ul>

# 2.2.9.4 Zjištění stavu pohybu chapadla

Prototyp	GetGripperMotionDone()
Popis	Zjištění stavu pohybu chapadla
Parametr	Nic
Návratová hodnota	<ul> <li>Úspěch:[0,status], status:0-nedokončený pohyb,1-ukončení cvičení</li> <li>Selhání:[errcode]</li> </ul>

# 2.2.9.5 Konfigurace chapadla

Prototyp	SetGripperConfig(company,device,softversion,bus)
Popis	Konfigurace chapadla
Parametr	<ul> <li>Společnost:Výrobci drápů, 1-Robotiq, 2-Huiling, 3-Tianji, 4-Dahuan, 5-Zhixing;</li> <li>zařízení:Číslo zařízení: Robotiq (řada 0-2F-85), Huiling (řada 0-NK, 1-Z-EFG-100), Tianji(0-TEG-110), Dahuan(0-PGI-140), Zhixing(0-CTPM2F20)</li> <li>softversion:Číslo verze softwaru, dočasně se nepoužívá, výchozí hodnota je 0;</li> <li>bus:Pozice sběrnice připojené k zařízení, dočasně se nepoužívá, výchozí hodnota je 0;</li> </ul>
Návratová hodnota	<ul> <li>Úspěch: [0]</li> <li>Selhání:[errcode]</li> </ul>

## 2.2.9.5.1 Příklad kódu

	ſ
1	import Mrpc
2	import time
3	#Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí ↔objekt robota
4	robot = Mrpc.RPC('192.168.58.2')
5	robot.SetGripperConfig(4,0,0,1) # Konfigurace upínacích klepet
6	time.sleep(1)
7	config = robot.GetGripperConfig() # získejte konfiguraci chapadla
8	print(config)
9	robot.ActGripper(1,0) # Claw reset
10	time.sleep(1)
11	robot.ActGripper(1,1) # <i>Aktivace drápu</i>
12	time.sleep(2)
13	robot.MoveGripper(1,100,48,46,30000,0) # <i>Pohyb drápů</i>
14	time.sleep(3)
15	robot.MoveGripper(1,0,50,0,30000,0) ret
16	= robot.GetGripperMotionDone() # <i>Příklad Dotaz na stav pohybu drápu</i>
17	print(ret)

# 2.2.10 Kontrola síly

# 2.2.10.1 Získání konfigurace snímače síly

Prototyp	FT_GetConfig()
Popis	Získání konfigurace snímače síly
Parametr	Nic
Návratová hodnota	<ul> <li>Úspěch:[0, společnost,zařízení,softverze,sběrnice],společnost:Výrobce senzoru</li> <li>Selhání:[errcode]</li> </ul>

## 2.2.10.2 Konfigurace snímače síly

Prototyp	FT_SetConfig(company,device,softversion,bus)
Popis	Konfigurace snímače síly
Parametr	<ul> <li>společnost:Výrobce senzorů,17-Kunwei Technology,19-Aerospace 11. institut,20-ATI senzory, 21-Zhongke Mi Dian, 22-Weihang Sensitive Core;</li> <li>číslo zařízení: Kunwei (0-KWR75B), Letecký a kosmický institut 11 (0- MCS6A-200-4), ATI (0-AXIA80-M8), Zhongkomi Point (0-MST2010), Wei- zavěsit Minxin (0-WHC6L-YB-10A);</li> <li>softversion:číslo verze softwaru, dočasně se nepoužívá, výchozí hodnota je 0;;</li> <li>bus:pozice sběrnice připojené k zařízení, dočasně se nepoužívá, výchozí hodnota je 0;</li> </ul>
Návratová hodnota	<ul> <li>Úspěch: [0]</li> <li>Selhání:[errcode]</li> </ul>
#### 2.2.10.2.1 Příklad kódu



#### 2.2.10.3 Aktivace snímače síly

Prototyp	FT_Activate(state)
Popis	Aktivace snímače síly
Parametr	<ul> <li>stav:0-Reset,1-Aktivovat</li> </ul>
Návratová hodnota	<ul><li>Úspěch: [0]</li><li>Selhání:[errcode]</li></ul>

#### 2.2.10.3.1 Příklad kódu

1	import Mrpc
2	#Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení <mark>vrátí</mark> ⊶objekt robota
3	robot = Mrpc.RPC('192.168.58.2')
4	robot.FT_Activate(0) #Resetování senzoru
5	time.sleep(1)
6	robot.FT_Activate(1) #Aktivace senzoru
7	time.sleep(1)

#### 2.2.10.4 Nulová kalibrace snímače síly

Prototyp	FT_SetZero(stav)
Popis	Nulová kalibrace snímače síly
Parametr	<ul> <li>stav:0-Odstranit nulu,1-Oprava nuly</li> </ul>
Návratová hodnota	<ul><li>Úspěch: [0]</li><li>Selhání:[errcode]</li></ul>

#### 2.2.10.4.1 Příklad kódu

- 1 import Mrpc
- ² # Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí ↔objekt robota
- <sup>3</sup> robot = Mrpc.RPC('192.168.58.2')
- <sup>4</sup> robot.FT\_SetZero(0) #Odstranění nuly ze senzoru
- ₅ time.sleep(1)
- <sup>6</sup> robot.FT\_SetZero(1) #Nulový bod senzoru by měl být opraven. Vezměte prosím na vědomí  $\rightarrow_{2e}$  na konci snímače nelze instalovat žádný nástroj.
- <sup>7</sup> time.sleep(1)

#### 2.2.10.5 Nastavení referenčního souřadného systému snímače síly

Prototyp	FT_SetRCS(ref)
Popis	Nastavení referenčního souřadného systému snímače síly
Parametr	• ref:0-Souřadnicový systém nástroje,1-Základní souřadnicový systém
Návratová hodnota	<ul> <li>Úspěch: [0]</li> <li>Selhání:[errcode]</li> </ul>

#### 2.2.10.5.1 Příklad kódu

1	import Mrpc
2	#Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí∟ → objekt robota
	·
3	robot = Mrpc.RPC('192.168.58.2')
4	robot.FT_SetRCS(0) #Nastavení referenčního souřadného systému na souřadný systém
	↔souřadnicový systémástroją.0základní
5	timě:sleep()ý systém

#### 2.2.10.6 Výpočet identifikace hmotnosti nákladu

Prototyp	FT_PdIdenCompute()
Popis	Výpočet identifikace hmotnosti nákladu
Parametr	Nic
Návratová hodnota	<ul><li>Success:[0,weight] ,weight-Load weight,unit[kg]</li><li>Selhání:[errcode]</li></ul>

#### 2.2.10.7 Záznam o identifikaci hmotnosti nákladu

Prototyp	FT_PdIdenRecord(tool_id)
Popis	Záznam o identifikaci hmotnosti nákladu
Parametr	<ul> <li>tool_id:číslo souřadnice senzoru, rozsah[0~14]</li> </ul>
Návratová hodnota	<ul><li>Úspěch: [0]</li><li>Selhání:[errcode]</li></ul>

#### 2.2.10.7.1 Příklad kódu

1	import Mrpc
2	import time
3	#Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí →objekt robota
4	robot = Mrpc.RPC('192.168.58.2')
5	#Identifikace zátěže. V tuto chvíli je nástroj, který má být identifikován, nainstalován na
6	konci. → Nástroj je nainstalován pod snímačem síly a jeho konec je svisle dolů. robouřednicené svistém mastavi základní souřadný systém na souřadný systém nástroje, 0- souradnicový system
7	time.sleep(1)
8	tool id = 10 #Souřadnicové číslo senzoru # Doloh a gríma če vzhlodom ko koncové
9	tool coord = $[0.0.0.35.0.0.0.0.0.0.0.0]$
10 11	$teel_treat = 0 #0 N hotistical e has a solution of the solut$
12	robot.SetToolCoord(tool_id,tool_coord,tool_type,tool_install) #Nastavení souřadnic
13	senzoru. time.sleep(1)
14	
15	time.sleep(1)
16	
17	weight = robot.F'I_PdIdenCompute() #Vypočítaná hmotnost zátěže, jednotka[kg]
	print(hmotnost)

#### 2.2.10.8 Výpočet identifikace centroidu zatížení

Prototyp	FT_PdCogIdenCompute()
Popis	Výpočet identifikace centroidu zatížení
Parametr	Nic
Návratová hodnota	<ul> <li>Úspěch:[0,cog],cog=[cogx,cogy,cogz] ,Load centroid,unit[mm]</li> <li>Selhání:[errcode]</li> </ul>

#### 2.2.10.9 Načtení identifikačního záznamu centroidu

Prototyp	FT_PdCogIdenRecord(tool_id)
Popis	Načtení identifikačního záznamu centroidu
Parametr	<ul> <li>tool_id:číslo souřadnice senzoru, rozsah[0~14]</li> </ul>
Návratová hodnota	<ul><li>Úspěch: [0]</li><li>Selhání:[errcode]</li></ul>

#### 2.2.10.9.1 Příklad kódu

1	import Mrpc
2	čas importu
3	# Je navázáno spojení s řídicí jednotkou robota. Při úspěšném spojení se vrátí
	$\leftrightarrow objekt robota$
4	robot = Mrpc.RPC('192.168.58.2')
5	#Pro identifikaci centroidu zátěže je třeba robota naučit tři různé polohy, pak → zaznamená identifikační údaje a nakonec vypočítá centroid zatížení.
6	P1=[-160.619,-586.138,384.988,-170.166,-44.782,169.295]
7	robot.MoveCart(P1,9,0,100.0,100.0,100.0,-1.0,-1)
8	time.sleep(1)
9	robot.FT_PdCogIdenRecord(tool_id,1) #Identifikace
	↔data záznamu_
10	time.sleep(1)
11	P2=[-8/.615,-606.209,556.119,-102.495,10.118,1/8.985]
12	robot.MoveCart(P2,9,0,100.0,100.0,100.0,-1.0,-1)
13	time.sleep(1)
14	time_sleep(1)
15	P3=[41,479,-557,243,484,407,-125,174,46,995,-132,165]
10	robot.MoveCart(P3,9,0,100.0,100.0,100.0,-1.0,-1)
18	time sleen(1)
19	robot.FT_PdCogIdenRecord(tool_id,3)
20	time.sleep(1)
21	cog = robot.FT_PdCogIdenCompute() # Vypočítaný a identifikovaný centroid zátěže
22	print(cog)

### 2.2.10.10 Získání údajů o síle/kroutícím momentu v referenčním souřadném systému

Prototyp	FT_GetForceTorqueRCS()
Popis	Získání údajů o síle/kroutícím momentu v referenčním souřadném systému
Parametr	Nic
Návratová hodnota	<ul> <li>Úspěch:[0,data] ,data=[fx,fy,fz,mx,my,mz]</li> <li>Selhání:[errcode]</li> </ul>

#### 2.2.10.10.1 Příklad kódu

1	import Mrpc
2	#Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení <mark>vrátí _</mark> ↔objekt robota
3	robot = Mrpc.RPC('192.168.58.2')
4	rcs = robot.FT_GetForceTorqueRCS() #Dotaz na data v souřadnicovém systému senzoru
5	print(rcs)

#### 2.2.10.11 Získání nezpracovaných údajů o síle/otáčivém momentu M om snímače síly

Prototyp	FT_GetForceTorqueOrigin()
Popis	Získání nezpracovaných údajů o síle/otáčivém momentu M om snímače síly
Parametr	Nic
Návratová hodnota	<ul> <li>Úspěch:[0,data] ,data=[fx,fy,fz,mx,my,mz]</li> <li>Selhání:[errcode]</li> </ul>

#### 2.2.10.11.1 Příklad kódu

1 import Mrpc

2

- # Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí \_ ↔objekt robota
- <sup>3</sup> robot = Mrpc.RPC('192.168.58.2')
- <sup>4</sup> origin = robot.FT\_GetForceTorqueOrigin() #*Příklad Dotaz na původní data senzoru*

₅ print(origin)

#### 2.2.10.12 Ochrana proti kolizi

Prototyp	FT_Guard(flag,sensor_num,select,force_torque,max_threshold, min_threshold)
Popis	Ochrana proti kolizi
Parametr	<ul> <li>příznak:0-Vypnout ochranu proti kolizi, 1-Zapnout ochranu proti kolizi;</li> <li>sensor_num:Číslo senzoru síly;</li> <li>vybrat:zda na šest stupňů na volnosti zjistit kolize[fx,fy,fz,mx,my,mz],0-neúčinné, 1-účinné;</li> <li>force_torque:Síla/moment detekce kolize, jednotka[N nebo Nm];</li> <li>max_threshold:Maximální prahová hodnota;</li> <li>min_threshold:Minimální prahová hodnota;</li> <li>Síla/kroutící moment detekce rozsah:(force_torque-min_threshold,force_torque+max_threshold)</li> </ul>
Návratová hodnota	<ul><li>Úspěch: [0]</li><li>Selhání:[errcode]</li></ul>

#### 2.2.10.12.1 Příklad kódu

import Mrpc 1 # Je navázáno spojení s řídicí jednotkou robota. Úspěšné spojení vrátí 2 *↔objekt robota* robot = Mrpc.RPC('192.168.58.2')3 #Enable flag, 0-Disable collision guard, 1-Enable collision guard actFlag = 14 sensor\_num = 1 #Číslo snímače síly 5 is\_select = [1,1,1,1,1,1,1] *#Zda Šest stupňů volnosti detekuje kolizi[fx,* 6 →ty,fz,mx,my,mz],0-neúčinné, 1-účinné force\_torque = [0.0,0.0,0.0,0.0,0.0,0.0,0.0] #Detekce síly/momentů kolize, detekce 7 ←rozsah sila\_moment-min\_prah, Sila moment+max prah max\_threshold = [10.0,10.0,10.0,10.0,10.0,10.0,10.0] #Maximální práh 8 min\_threshold = [5.0,5.0,5.0,5.0,5.0,5.0,5.0,5.0] #Minimální práh 9 P1=[-160.619,-586.138,384.988,-170.166,-44.782,169.295] 10 P2=[-87.615,-606.209,556.119,-102.495,10.118,178.985] 11 P3=[41.479,-557.243,484.407,-125.174,46.995,-132.165] 12 robot.FT\_Guard(actFlag, sensor\_num, is\_select, force\_torque, max\_threshold, min\_ 13 #Zapnout ochranu proti kolizím *⊶prahová* hodnota robot.MoveCart(P1,9,0,100.0,100.0,100.0,-1.0,-1) *#Pohyb z bodu do bodu v* 14 kloubu  $\leftrightarrow prostor$ 15 robot.MoveCart(P2,9,0,100.0,100.0,100.0,-1.0,-1) 16 robot.MoveCart(P3,9,0,100.0,100.0,100.0,-1.0,-1) 17 actFlag = 018 robot.FT\_Guard(actFlag, sensor\_num, is\_select, force\_torque, max\_threshold, min\_ #Zakázat ochranu proti kolizím →prahová hodnota)

#### 2.2.10.13 Řízení konstantní síly

Prototyp Popis	FT_Control(flag,sensor_num,select,force_torque,gain,adj_sign, ILC_sign,max_dis,max_ang) Řízení konstantní síly
Parametr	<ul> <li>flag:Příznak otevřeného řízení konstantní síly, 0-off, 1-on;</li> <li>sensor_num:Číslo senzoru síly;</li> <li>vybrat:Je zjištěno šest stupňů volnosti [fx,fy,fz,mx,my,mz],0- neúčinné, 1-účinné;</li> <li>force_torque:Detekční síla/moment, jednotka[N nebo Nm];</li> <li>gain:[f_p,f_i,f_d,m_p,m_i,m_d],Force PID parameters, Torque PID parameters;</li> <li>adj_sign:Adaptive start stop status, 0-off, 1-on;</li> <li>ILC_sign: 0-stop, 1-trénink, 2-praktický provoz;</li> <li>max_dis:Maximální vzdálenost nastavení;</li> <li>max_ang:Maximální úhel nastavení;</li> </ul>
Návratová hodnota	<ul><li>Úspěch: [0]</li><li>Selhání:[errcode]</li></ul>

# 2.2.10.13.1 Příklad kódu

1	import Mrpc
2	#Je navázáno spojení s řídicí jednotkou robota. Při úspěšném spojení se vrátí ↔objekt robota
3	robot = Mrpc.RPC('192.168.58.2')
4	status = 1 #Příznak otevření řízení konstantní síly, 0-off, 1-on
5	sensor_num = 1 #Císlo senzoru síly
6	is_select = [0,0,1,0,0,0] #Šest stupňů volnosti choice[fx,fy,fz,mx,my,mz],0- →neúčinné, 1-účinné
7	force_torque = [0.0,0.0,-10.0,0.0,0.0,0.0,0.0] #Síla a točivý moment pro detekci kolize, ↔ rozsah detekce síla_moment-min_prah,síla_moment+max_prah
8	gain = [0.0005,0.0,0.0,0.0,0.0,0.0] #Maximální práh
9	adj_sign = 0 #Adaptive start stop status, 0-off, 1-on
10	ILC_sign = 0 #ILC control start stop status, 0-stop, 1-trénink, 2-praktický provoz
11	max_dis = 100.0 #Maximální vzdálenost nastavení
12	max_ang = 0,0 #Maximální úhel nastavení
13	J1=[-68.987,-96.414,-111.45,-61.105,92.884,11.089]
14	P1=[62.795,-511.979,291.697,-179.545,3.027,-170.039]
15	eP1=[0.000,0.000,0.000]
16	dP1=[0.000,0.000,0.000,0.000,0.000]
17	J2=[-107.596,-109.154,-104.735,-56.176,90.739,11.091]
18	P2=[-294.768,-503.708,233.158,179.799,0.713,151.309]
19	eP2=[0.000,0.000,0.000]
20	dP2=[0.000,0.000,0.000,0.000,0.000]
21	robot.MoveJ(J1,P1,9,0,100.0,180.0,100.0,eP1,-1.0,0,dP1) #Pohyb v kloubovém prostoru PTP, ↔ nástroj číslo 9, skutečný test byl použit podle údajů z terénu a čísla nástroje
22	robot.FT_Control(status,sensor_num,is_select,force_torque,gain,adj_sign,ILC_sign,max_dis,
	,→max_ang) #Řízení konstantní silou
23	robot.MoveL(J2,P2,9,0,100.0,180.0,20.0,-1.0,eP2,0,0,dP2) #Rectilineární pohyb $v_{\_}$ $↔$ Kartézský prostor
24	status = 0
25	robot.FT_Control(status,sensor_num,is_select,force_torque,gain,adj_sign,ILC_sign,max_dis,
	(→max_ang)

### 2.2.10.14 Průzkum spirálovitých linií

Prototyp	FT_SpiralSearch(rcs,dr,fFinsih,t,vmax)
Popis	Průzkum spirálovitých linií
Parametr	<ul> <li>rcs:Referenční souřadnicový systém, souřadnicový systém 0, souřadnicový systém 1-základna</li> <li>dr:Rychlost posuvu na poloměr kružnice, jednotka[mm];</li> <li>fFinish:Práh síly nebo točivého momentu (0-100), jednotka[N/Nm];</li> <li>t:Maximální doba průzkumu, jednotka[ms];</li> <li>vmax: Maximální lineární rychlost, jednotka [mm/s]</li> </ul>
Návratová hodnota	<ul> <li>Úspěch: [0]</li> <li>Selhání:[errcode]</li> </ul>

# 2.2.10.14.1 Příklad kódu

1	import Mrpc
2	#Je navázáno spojení s řídicí jednotkou robota. Při úspěšném spojení se vrátí
	$\leftrightarrow objekt robota$
3	robot = Mrpc.RPC('192.168.58.2')
4	#Konstantnl parametr slly
5	status = 1 #Příznak otevření řízení konstantní síly, 0-off, 1-on
6	sensor_num = 1 #Císlo senzoru síly
7	is_select = $[0,0,1,0,0,0]$ #Šest stupňů volnosti choice[fx,fy,fz,mx,my,mz],0- $\rightarrow$ neúčinné, $1-$ účinné
8	force_torque = $[0.0,0.0,-10.0,0.0,0.0,0.0]$ #Síla a točivý moment pro detekci kolize, $\leftrightarrow$ rozsah detekce síla_moment-min_prah,síla_moment+max_prah
9	gain = [0.0001,0.0,0.0,0.0,0.0,0.0] #Maximální práh
10	adj_sign = 0 #Adaptive start stop status, 0-off, 1-on
11	ILC_sign = 0 #ILC control start stop status, 0-stop, 1-trénink, 2-praktický provoz
12	max_dis = 100.0 #Maximální vzdálenost nastavení
13	max_ang = 5.0 #Maximální úhel nastavení
14	#Helix prozkoumat parametry
15	rcs = 0 #Reference M ame, 0-Tool M ame, 1-Base M ame
16	dr = 0,7 #Podávání na poloměr kruhu, jednotka[mm]
17	fFinish = 1.0 #Prahová hodnota síly nebo momentu0~100,jednotka[N nebo Nm]
18	t = 60000.0 #Maximální doba průzkumu, jednotka[ms]
19	vmax = 3.0 #Maximální lineární rychlost, jednotka[mm/s]
20	is_select = $[0,0,1,1,1,0]$ #Šest stupňů volnosti choice[fx,fy,fz,mx,my,mz],0- $\rightarrow$ neúčinné, $1-$ účinné
21	robot.FT_Control(status,sensor_num,is_select,force_torque,gain,adj_sign,ILC_sign,max_dis,
	( →max_ang)
22	robot.FT_SpiralSearch(rcs,dr,fFinish,t,vmax)
23	status = 0
24	robot.FT_Control(status,sensor_num,is_select,force_torque,gain,adj_sign,ILC_sign,max_dis,
	(max_ang)

## 2.2.10.15 Otáčení vložky

Prototyp	FT_RotInsertion(rcs,angVelRot,forceInsertion,angleMax,orn, angAccmax rotorn)
Popis	Otáčení vložky
Parametr	<ul> <li>rcs:Referenční souřadnicový systém, souřadnicový systém 0, souřadnicový systém 1;</li> <li>angVelRot:Úhlová rychlost otáčení: uni[t°/s];</li> <li>forceInsertion:Práh síly nebo točivého momentu (0~100), jednotka[N nebo Nm];</li> <li>angleMax:maximální úhel natočení, jednotka[°];</li> <li>orn:Směr síly, 1-fz,2-mz;</li> <li>angAccmax:Maximální rotační zrychlení, jednotka[°/s^2],dočasně se nepoužívá</li> <li>rotorn:Směr otáčení, 1 ve směru hodinových ručiček, 2 proti směru hodinových ručiček</li> </ul>
Návratová hodnota	<ul> <li>Úspěch: [0]</li> <li>Selhání:[errcode]</li> </ul>

### 2.2.10.15.1 Příklad kódu

1	import Mrpc
2	#Je navázáno spojení s řídicí jednotkou robota. Při úspěšném spojení se vrátí ↔objekt robota
3 4	robot = Mrpc.RPC('192.168.58.2') #Konstantní parametr síly
5	status = 1 #Příznak otevření řízení konstantní síly, 0-off, 1-on
6	sensor_num = 1 #Číslo senzoru síly
7	is_select = $[0,0,1,0,0,0]$ #Šest stupňů volnosti choice[fx,fy,fz,mx,my,mz],0- $\rightarrow$ neúčinné, $1-účinné$
8	force_torque = [0.0,0.0,-10.0,0.0,0.0,0.0,0.0] #Síla a točivý moment pro detekci kolize, ↔ rozsah detekce síla_moment-min_prah,síla_moment+max_prah
9	gain = [0.0001,0.0,0.0,0.0,0.0,0.0] #Maximální práh
10	adj_sign = 0 #Adaptive start stop status, 0-off, 1-on
11	ILC_sign = 0 #ILC control start stop status, 0-stop, 1-trénink, 2-praktický provoz
12	max_dis = 100.0 #Maximální vzdálenost nastavení
13 14	max_ang = 5.0 #Maximální úhel nastavení #Rotační parametr vložení
15	rcs = 0 #Reference M ame, 0-Tool M ame, 1-Base M ame
16	angVelRot = 2,0 #Rotační úhlová rychlost, jednotka[°/s]
17	forceInsertion = 1.0 #Prahová hodnota síly nebo momentu0~100, jednotka[N nebo Nm]
18	angleMax= 45 #Maximální úhel natočení, jednotka[°]
19	orn = 1 #Smersily, 1-fz, 2-mz
20	angAccmax = 0.0 #Maximální rotační zrychlení, jednotka [°/s^2], dočasně se nepoužívá
21	rotorn = 1 #Smer otacent, 1 proti smeru hodinových ručíček, 2 proti smeru hodinových ručíček
22	s_select = [0,0,1,1,1,0] #Sest stupňů volnosti choice[fx,fy,fz,mx,my,mz],0- →neúčinné, I-účinné
23	force_torque = [0.0,0.0,-10.0,0.0,0.0,0.0,0.0] #Síla a točivý moment pro detekci kolize, ↔ rozsah detekce síla_moment-min_prah,síla_moment+max_prah
24	gain = [0.0001,0.0,0.0,0.0,0.0,0.0] #Maximální práh
25	status = 1
26	robot.FT_Control(status,sensor_num,is_select,force_torque,gain,adj_sign,ILC_sign,max_dis,
	·→max_ang)
27	robot.FT_RotInsertion(rcs,angVelRot,forceInsertion,angleMax,orn,angAccmax,rotorn)
28	status = 0
29	robot.F'l'_Control(status,sensor_num,is_select,force_torque,gain,adj_sign,ILC_sign,max_dis,
	← max_ang)

#### 2.2.10.16 Lineární vkládání

Prototyp Popis	FT_LinInsertion(rcs,force_goal,lin_v,lin_a,disMax,linorn) Lineární vkládání
Parametr	<ul> <li>rcs:Reference M ame, 0-Tool M ame, 1-Base M ame;</li> <li>force_goal:Prahová hodnota síly nebo točivého momentu, jednotka[N nebo Nm];</li> <li>lin_v:Lineární rychlost, jednotka[mm/s];</li> <li>lin_a:Lineární zrychlení, jednotka[mm/s^2],dočasně nepoužíváno;</li> <li>disMax: Maximální vzdálenost vložení, jednotka [mm];</li> </ul>
	<ul> <li>linorn:Směr vložení, 1 - kladný směr, 2 - záporný směr;</li> </ul>
Návratová hodnota	<ul> <li>Úspěch: [0]</li> <li>Selhání:[errcode]</li> </ul>

# 2.2.10.16.1 Příklad kódu

1	import Mrpc
2	#Je navázáno spojení s řídicí jednotkou robota. Při úspěšném spojení se <mark>vrátí</mark> ↔objekt robota
3 4	robot = Mrpc.RPC('192.168.58.2') #Konstantní parametr síly
5	status = 1     #Příznak otevření řízení konstantní síly, 0-off, 1-on
6	sensor_num = 1 #Číslo senzoru síly
7	is_select = $[0,0,1,0,0,0]$ #Šest stupňů volnosti choice[fx,fy,fz,mx,my,mz],0- $\rightarrow$ neúčinné, $1-$ účinné
8	force_torque = [0.0,0.0,-10.0,0.0,0.0,0.0,0.0] #Síla a točivý moment pro detekci kolize, ↔ rozsah detekce síla_moment-min_prah,síla_moment+max_prah
9	gain = [0.0001,0.0,0.0,0.0,0.0,0.0] #Maximální práh
10	adj_sign = 0 #Adaptive start stop status, 0-off, 1-on
11	ILC_sign = 0 #ILC control start stop status, 0-stop, 1-trénink, 2-praktický provoz
12	max_dis = 100.0 #Maximální vzdálenost nastavení
13 14	max_ang = 5.0 #Maximální úhel nastavení #Lineární parametr vložení
15	rcs = 0 #Reference M ame, 0-Tool M ame, 1-Base M ame
16	force_goal = 20.0 #Prahová hodnota síly nebo momentu0~100, jednotka[N nebo Nm]
17	lin_v = 0.0 #Lineární rychlost, jednotka[mm/s]
18	lin_a = 0.0 #Lineární zrychlení, jednotka[mm/s^2],dočasně nepoužívané
19	disMax = 100.0 #Maximální vzdálenost vložení, jednotka[mm]
20	linorn = 1 # Směr vložení, 1-pozitivní směr, 2-negativní směr
21	is_select = $[1,1,1,0,0,0]$ #Šest stupňů volnosti choice[fx,fy,fz,mx,my,mz],0- $\rightarrow$ neúčinné, $1-$ účinné
22	gain = [0.00005,0.0,0.0,0.0,0.0,0.0] #Maximální práh
23	force_torque = [0.0,0.0,-30.0,0.0,0.0,0.0,0.0] #Síla a točivý moment pro detekci kolize, ↔ rozsah detekce síla_moment-min_prah,síla_moment+max_prah
24	status = 1
25	robot.FT_Control(status,sensor_num,is_select,force_torque,gain,adj_sign,ILC_sign,max_dis,
	⇔max_ang)
26	robot.FT_LinInsertion(rcs,force_goal,lin_v,lin_a,disMax,linorn)
	(pokračování na další straně)

(pokračování na předchozí straně)

- status = 0

#### 2.2.10.17 Výpočet polohy střední roviny pro začátek

Prototyp	FT_CalCenterStart()
Popis	Výpočet polohy střední roviny pro začátek
Parametr	Nic
Návratová hodnota	<ul> <li>Úspěch: [0]</li> <li>Selhání:[errcode]</li> </ul>

### 2.2.10.18 Výpočet polohy střední roviny na konci

Prototyp	FT_CalCenterEnd()
Popis	Výpočet polohy střední roviny na konci
Parametr	Nic
Návratová hodnota	<ul> <li>Úspěch:[0,pos] ,pos=[x,y,z,rx,ry,rz]</li> <li>Selhání:[errcode]</li> </ul>

#### 2.2.10.19 Polohování na povrchu

Prototyp	FT_FindSurface (rcs,dir,axis,lin_v,lin_a,disMax,force_goal)
Popis	Polohování na povrchu
Parametr	<ul> <li>rcs: Referenční název, 0-Nástrojový název, 1-Bázový název;</li> <li>dir:Směr pohybu, 1-pozitivní, 2-negativní;</li> <li>axis:Move Axis,1-x,2-y,3-z;</li> <li>lin_v:Zkoumání lineární rychlosti, jednotka[mm/s];</li> <li>lin_a:Zkoumání lineárního zrychlení, jednotka[mm/s^2];</li> <li>disMax:Maximální vzdálenost průzkumu, jednotka[mm]</li> <li>force_goal:Action termination force threshold,unit[N];</li> </ul>
Návratová hodnota	<ul><li>Úspěch: [0]</li><li>Selhání:[errcode]</li></ul>

#### 2.2.10.19.1 Příklad kódu

1	import Mrpc
2	#Je navázáno spojení s řídicí jednotkou robota. Při úspěšném spojení se vrátí ↔objekt robota
3 4	robot = Mrpc.RPC('192.168.58.2') #Konstantní parametr síly
5	status = 1 #Příznak otevření řízení konstantní síly, 0-off, 1-on
6	sensor_num = 1 #Číslo senzoru síly
7	is_select = $[1,0,0,0,0,0,0]$ #Šest stupňů volnosti choice[fx,fy,fz,mx,my,mz],0- $\rightarrow$ neúčinné, $1-$ účinné
8	force_torque = [-2.0,0.0,0.0,0.0,0.0,0.0] #Síla a točivý moment pro detekci kolize, ↔ rozsah detekce síla_moment-min_prah,síla_moment+max_prah
9	gain = [0.0002, 0.0, 0.0, 0.0, 0.0, 0.0] #Maximální práh
10	adj_sign = 0 #Adaptive start stop status, 0-off, 1-on
11	ILC_sign = 0 #ILC control start stop status, 0-stop, 1-trénink, 2-praktický provoz
12	max_dis = 100.0 #Maximální vzdálenost nastavení
13 14	max_ang = 5.0 #Maximální úhel nastavení #Parametr polohování povrchu
15	rcs = 0 #Reference M ame, 0-Tool M ame, 1-Base M ame
16	direction = 1 #Směr pohybu, 1-pozitivní směr, 2-negativní směr
17	$axis = 1 \# Osa \ pohybu, 1-X, 2-Y, 3-Z$
18	$\lim_{x \to \infty} v = 3,0  \#Zkoumáni přímočaré rychlosti, jednotka[mm/s]$
19	$lin_a = 0,0  #Linearni zrychleni, jednotka[mm/s^2]$
20	disMax = 50.0 #Maximalni vzdalenost pruzkumu, jednotka[mm]
21 22	$force_goal = 2.0 \ #Prahova \ hodnota \ stly \ ukonCent \ akce, \ jednotka[N] \\ P1=[-230.959, -364.017, 226.179, -179.004, 0.002, 89.999]$
23	robot.MoveCart(P1,9,0,100.0,100.0,100.0,-1.0,-1)  # <i>Pohyb z bodu do bodu v</i> kloubu <i>← prostor</i>
24	#Hledejte střed ve směru x
25	#První povrch
26	robot.FT_CalCenterStart()
27	robot.FT_Control(status,sensor_num,is_select,force_torque,gain,adj_sign,ILC_sign,max_dis, 
28	robot.FT_FindSurface(rcs,direction,axis,lin_v,lin_a,disMax,force_goal)
29	status = 0
30	robot.FT_Control(status,sensor_num,is_select,force_torque,gain,adj_sign,ILC_sign,max_dis, →max_ang)
31	robot.MoveCart(P1,9,0,100.0,100.0,100.0,-1.0,-1) #Pohyb z bodu do bodu v kloubu_ ↔ prostor
32 33	robot.WaitMs(1000) <i>#Druhý povrch</i>
34	robot.FT_Control(status,sensor_num,is_select,force_torque,gain,adj_sign,ILC_sign,max_dis,
	→max_ang)
35	direction = 2 #Směr pohybu, 1-pozitivní směr, 2-negativní směr
36	robot.FT_FindSurface(rcs,direction,axis,lin_v,lin_a,disMax,force_goal)
37	status = 0
38	robot.FT_Control(status,sensor_num,is_select,force_torque,gain,adj_sign,ILC_sign,max_dis,
39	<sup>→mæ</sup> s <sup>ang</sup> #Vypočítejte polohu středu ve směru x
40	<pre>xcenter= robot.FT_CalCenterEnd()</pre>
41	print(xcenter)
42	xcenter = [ xcenter[1],xcenter[2],xcenter[3],xcenter[4],xcenter[5],xcenter[6]].

(pokračování na další straně)

(pokračování na předchozí straně)

- 43 robot.MoveCart(xcenter,9,0,60.0,50.0,50.0,0.0,-1)
- 44 #Hledejte střed ve směru y
- 45 #První povrch
- <sup>46</sup> robot.FT\_CalCenterStart()
- <sup>47</sup> robot.FT\_Control(status,sensor\_num,is\_select,force\_torque,gain,adj\_sign,ILC\_sign,max\_dis, →max\_ang)
- 48 direction = 1 #Směr pohybu, 1-pozitivní směr, 2-negativní směr
- 49 axis = 2 #Osa pohybu, 1-X, 2-Y, 3-Z
- <sup>50</sup> disMax = 150.0 #Maximální vzdálenost průzkumu, jednotka[mm]
- $lin_v = 6,0 \quad #Zkoumání přímočaré rychlosti, jednotka[mm/s]$
- <sup>52</sup> robot.FT\_FindSurface(rcs,direction,axis,lin\_v,lin\_a,disMax,force\_goal)
- status = 0
- <sup>54</sup> robot.FT\_Control(status,sensor\_num,is\_select,force\_torque,gain,adj\_sign,ILC\_sign,max\_dis, →max\_ang)
- robot.MoveCart(P1,9,0,100.0,100.0,100.0,-1.0,-1) #Pohyb z bodu do bodu v kloubu  $\rightarrow$  prostor
- <sup>56</sup> robot.WaitMs(1000)
- 57 #Druhý povrch
- <sup>58</sup> robot.FT\_Control(status,sensor\_num,is\_select,force\_torque,gain,adj\_sign,ILC\_sign,max\_dis, →max\_ang)
- <sup>59</sup> direction = 2 #Směr pohybu, 1-pozitivní směr, 2-negativní směr
- <sup>60</sup> robot.FT\_FindSurface(rcs,direction,axis,lin\_v,lin\_a,disMax,force\_goal)
- $_{61}$  status = 0
- <sup>62</sup> robot.FT\_Control(status,sensor\_num,is\_select,force\_torque,gain,adj\_sign,ILC\_sign,max\_dis,
- G3 #Vypočítejte středovou polohu y
- <sup>64</sup> ycenter=robot.FT\_CalCenterEnd()
- 65 print(ycenter)
- <sup>66</sup> ycenter = [ ycenter[1], ycenter[2], ycenter[3], ycenter[4], ycenter[5], ycenter[6]]
- <sup>67</sup> robot.MoveCart(ycenter,9,0,60.0,50.0,50.0,-1.0,-1)

#### 2.2.10.20 Vypnutá kontrola flexibility

Prototyp	FT_ComplianceStop()
Popis	Vypnutá kontrola flexibility
Parametr	Nic
Návratová hodnota	<ul><li>Úspěch: [0]</li><li>Selhání:[errcode]</li></ul>

### 2.2.10.21 Kontrola flexibility na

Prototyp	FT_ComplianceStart(p,force)
Popis	Kontrola flexibility na
Parametr	<ul> <li>p: Koeficient nastavení polohy nebo koeficient shody</li> <li>force:flexibilita práh otevírací síly, jednotka[N]</li> </ul>
Návratová hodnota	<ul><li>Úspěch: [0]</li><li>Selhání:[errcode]</li></ul>

#### 2.2.10.21.1 Příklad kódu

1	import Mrpc
2	#Je navázáno spojení s řídicí jednotkou robota. Při úspěšném spojení se vrátí∟ ⊶objekt robota
3	robot = Mrpc.RPC('192.168.58.2')
4	J1=[-105.3,-68.0,-127.9,-75.5,90.8,77.8]
5	P1=[-208.9,-274.5,334.6,178.8,-1.3,86.7]
6	eP1=[0.000,0.000,0.000]
7	dP1=[0.000,0.000,0.000,0.000,0.000]
8	]2=[-105.3,-97.9,-101.5,-70.3,90.8,77.8]
9	P2=[-264.8,-480.5,341.8,179.2,0.3,86.7]
10	eP2=[0.000,0.000,0.000]
11 12	dP2=[0.000,0.000,0.000,0.000,0.000] #Konstantni parametr síly
13	status = 1 #Constant force control open flag, 0-off, 1-on
14	sensor_num = 1 #Číslo senzoru síly
15	is_select = [1,0,0,0,0,0,0] #Šest stupňů volnosti choice[fx,fy,fz,mx,my,mz],0- →neúčinné, 1-účinné
16	$force\_torque = [-2.0, 0.0, 0.0, 0.0, 0.0, 0.0] #Sila a moment detekce kolize, \_ (\rightarrow rozsah_{detekce} sila\_moment-min\_prah, sila\_moment+max\_prah]$
17	gain = [0.0002,0.0,0.0,0.0,0.0,0.0] #Maximální práh
18	adj_sign = 0 #Adaptive start stop status, 0-off, 1-on
19	ILC_sign = 0 #ILC control start stop status, 0-stop, 1-trénink, 2-praktický provoz
20	max_dis = 100.0 #Maximální vzdálenost nastavení
21	max_ang = 5.0 #Maximální úhel nastavení
22	robot FT Control(status sonsor num is soloet force torque gain adi sign U.C. sign max dis
23	$10000.111_00010000000000000000000000000$
24	$m_{\text{max}_{\text{ang}}}$ $\mathbf{n} = 0.00005$ #Kaeficient nřiznůsobení poloby nebo skody
24	force = 30.0  #Prabová hodnota sílv při otevření jednotka[N]
25	robot FT_ComplianceStart(p_force)
20	count = 15
28	while(count):
20	robot Movel (I1 P1 9 0 100 0 180 0 100 0 -1 0 eP1 0 1 dP1) $\#$ Rectilineární pohyby
29	↔Kartézský prostor
30	robot.MoveL(J2,P2,9,0,100.0,180.0,100.0,-1.0,eP2,0,0,dP2)
31	count = count - 1
	(pokračování na další straně)

(pokračování na předchozí straně)

- robot.FT\_ComplianceStop() status = 0 32
- 33
- robot.FT\_Control(status,sensor\_num,is\_select,force\_torque,gain,adj\_sign,ILC\_sign,max\_dis, 34

# 2.3 Srovnávací tabulka chybových kódů

Errcode	Popište	Způsob zpracování
-1	Další chyby	Kontaktujte poprodejního technika, abyste si mohli prohlédnout protokol řídicí jednotky.
0	Úspěšné volání	1
3	Počet parametrů rozhraní je nekonzistentní	Kontrola počtu parametrů rozhraní
4	Výjimka hodnoty parametru rozhraní	Zkontrolujte typ nebo rozsah hodnoty parametru
8	Nepodařilo se otevřít soubor se stopou	Zkontrolujte, zda soubor stopy TPD existuje nebo zda je název stopy správně
14	Provedení rozhraní se nezdařilo	Zkontrolujte, zda webové rozhraní hlásí závadu nebo stav. zpětná vazba hlásí závadu
18	Program robota je spuštěn, prosím, zastavte to jako první	Zastavení programu před provedením dalších operací
25	Datová výjimka, výpočet se nezdařil	Rekalibrace nebo identifikace
28	Výpočet inverzní kinematiky re- výsledky jsou abnormální	Zkontrolujte, zda je póza přiměřená
29	Překročení kloubu ServoJ	Zkontrolujte, zda jsou společné údaje v přiměřeném rozmezí.
30	Neodstranitelná porucha, prosím, napájení vypnout a restartovat řídicí jednotku	Vypněte a restartujte řídicí jednotku
34	Špatné číslo obrobku	Zkontrolujte, zda je číslo obrobku přiměřené
36	Příliš dlouhý název souboru	Zkraťte prosím délku názvu souboru
38	Singulární póza, výpočet se nezdařil	Změňte prosím pózu
64	Nepřidává se do fronty pokynů	Kontaktujte poprodejního technika, abyste si mohli prohlédnout protokol řídicí jednotky.
66	Prostřední bod 1 celého okruhu příkaz cle/helix je chybný	Zkontrolujte, zda jsou údaje středního bodu 1 správné
67	Střední bod 2 plného okruhu příkaz cle/helix je chybný	Zkontrolujte, zda jsou údaje středního bodu 2 správné
68	Střední bod 3 plného okruhu příkaz cle/helix je chybný	Zkontrolujte, zda jsou údaje středního bodu 3 správné
69	Střední bod oblouku kom- mand se mýlí	Zkontrolujte, zda jsou údaje mezilehlého bodu správné
70	Chyba cílového bodu instrukce Arc	Zkontrolujte, zda jsou údaje o cílovém bodu správné
73	Chyba pohybu chapadla	Zkontrolujte, zda je stav komunikace chapadla je normální
74	Chyba řádkového bodu instrukce	Zkontrolujte, zda jsou údaje o bodech správné
75	Chyba kanálu	Zkontrolujte, zda je číslo IO v rozsahu
76	Časový limit čekání	Zkontrolujte, zda je vstupní signál IO nebo zda je zapojení správné. rect
82	Chyba bodu instrukce TPD	Znovu nahrát výukovou stopu
83	Nástroj pro instrukce TPD neodpovídá aktuální nástroj	Změna souřadnicového systému nástroje používaného při výuce na TPD
94	Chyba záchytného bodu Spline	Zkontrolujte, zda jsou údaje o bodech správné
108	Špatný výchozí bod pro helix com- mand	Zkontrolujte, zda jsou údaje o výchozím bodě správné
112	Danou pozici nelze dosáhnout	Zkontrolujte, zda je cílová póza přiměřená

# KAPITOLA

# MCOBOT\_ROS

# 3.1 Přehled

Stručná architektura frcobot\_ros je znázorněna na obrázku níže. Na straně kolaborativního robota je k dispozici server XMLRPC a server TCP.

- Server XMLRPC poskytuje především rozhraní API příkazů robota pro dokončení funkce získávání hodnot pohybu a stavu robota, které je založeno především na SDK C++.
- Server TCP zpětné vazby o stavu poskytuje zpětnou vazbu o stavu robota v reálném čase a perioda zpětné vazby je 8 ms.

ROS a Moveit! byly nainstalovány do počítače uživatele a byl zkompilován M cobot\_ros. Každý balík funkcí v M cobot\_ros obsahuje knihovnu lib API robota a v M cobot\_hw vytváří klienta TCP, který komunikuje se serverem zpětné vazby o stavu robota a získává data zpětné vazby o stavu robota.



# 3.2 Instalace

V této kapitole se dozvíte, jak sestavit M cobot\_ros a jak vytvořit potřebné instalační prostředí.

#### 3.2.1 Požadavky na životní prostředí

Doporučené prostředí pro M cobot\_ros je následující:

#### Poznámka:

- Ubuntu 18.04 LTS Bionic Beaver & ROS Melodic Morenia
- Ubuntu 20.04 LTS Focal Fossa a ROS Noetic Ninjemys

Níže uvedené pokyny jsou určeny pro systémy Ubuntu 20.04 LTS a ROS Noetic Ninjemys. Pokud používáte Melodic, nahraďte noetic v příkazovém řádku pomocí melodic.

#### 3.2.2 Instalace a požadavky na ROS

Po instalaci systému Ubuntu nainstalujte a nakonfigurujte prostředí ROS Noetic. Po

konfiguraci ROS Noetic nainstalujte požadované prostředí následujícím způsobem:

```
1 echo "source /opt/ros/noetic/setup.bash" >> ~/.bashrc
2 source ~/.bashrc
3 sudo apt-get install -y \
4 ros-noetic-rosparam-shortcuts \
5 ros-noetic-ros-control \
6 ros-noetic-ros-controllers \
7 ros-noetic-moveit
```

#### 3.2.3 Kompilace

Po správné instalaci a konfiguraci systému ROS Noetic vytvořte pracovní prostor Catkin ve zvoleném adresáři.

```
1 mkdir -p ~/catkin_ws/src
```

```
<sup>2</sup> cd ~/catkin_ws
```

```
<sup>3</sup> catkin_init_workspace src
```

Pak naklonujte knihovnu M cobot\_ros M om Github.

```
1 cd src
```

```
<sup>2</sup> git clone https://github.com/M-INNOVATION/Mcobot_ros.git
```

#### Sestavení balíčku M cobot\_ros

- 1 cd ~/catkin\_ws
- 2 catkin\_make
- 3 echo "source ~/catkin\_ws/devel/setup.bash" >> ~/.bashrc
- source ~/.bashrc

Pokud dojde k chybě, zkontrolujte, zda byly balíčky v požadavcích na instalaci ROS úspěšně nainstalovány. Po dokončení kompilace zkopírujte knihovnu lib do prostředí ROS lib (cesta je: /opt/ros/noetic/lib), aby program mohl normálně běžet.

- # Výchozí cesta catkin\_ws je zde "~", pokud je jiná, stačí změnit "~" na ↔ skutečná cesta
- <sup>2</sup> sudo cp ~/catkin\_ws/src/Mcobot\_ros/Mcobot\_hw/lib/\* /opt/ros/noetic/lib

# 3.3 Rychlý start

### 3.3.1 Mcobot\_hw

Mcobot\_hw poskytuje především základní funkce pro komunikaci s kolaborativními roboty.

#### Poznámka:

- Obsahuje zpětnou vazbu o stavu kolaborativního robota msg
- Poskytnout ukázky příkazů pro ovládání kolaborativních robotů.
- Poskytování zpětné vazby o stavu robotů pro spolupráci v uzlech a tématech
- Stavový uzel a demo příkazů lze rychle spustit pomocí spouštěcího souboru

Obsah souboru Mcobot\_hw.launch je následující:

```
<start>
1
2
        <!-- params -->
3
        <param name="robot_ip" type="string" value="192.168.58.2"/>
4
        <param name="robot_port" type="int" value="8083"/>
5
6
        <!-- Stavový uzel M cobot -->
7
        <node pkg="Mcobot_hw" type="Mcobot_status_node" name="Mcobot_status_node" output=
8
    , obrazovka" />
9
        <!-- M cobot control demo -->
10
        <node pkg="Mcobot_hw" type="Mcobot_cmd_demo" name="Mcobot_cmd_demo" output="screen
11
    .→" />
12
   </launch>
13
```

#### Důležité:

- robot\_ip a robot\_port musí odpovídat IP a portu ovládaného kolaborativního robota.
- Výchozí IP továrního robota je 192.168.58.2 a port zpětné vazby uživatele je 8083.

Následující příkazy slouží k rychlému spuštění uzlu zpětné vazby o stavu robota a funkce demonstrace příkazů.

<sup>1</sup> roslaunch Mcobot\_hw Mcobot\_hw.launch

Otevřete nový terminál a pomocí následujících příkazů vytiskněte a zobrazte zpětná data o stavu v reálném čase.

<sup>1</sup> rostopic ehco /Mcobot\_status

# KAPITOLA ČTYŘI

# MCOBOT\_ROS2

# 4.1 Přehled

Mcobot\_ros2 je rozhraní API vyvinuté kolaborativním robotem Fao na základě ROS2, jehož cílem je pohodlnější používání Fao SDK pro základní uživatele. Konfigurace výchozích parametrů prostřednictvím konfiguračního souboru parametrů se může přizpůsobit různým požadavkům zákazníků.

# 4.2 M\_ros2

Tato kapitola popisuje, jak nakonfigurovat běhové prostředí APP.

### 4.2.1 Základní prostředí instalace

Doporučujeme jej používat v systému Ubuntu22.04LTS (Jammy). Po instalaci systému můžete nainstalovat ROS2. Doporučuje se použít ros2-humble. Instalaci oMOS2 naleznete v návodu: h t t p s : //docs.ros.org/en/ humble/index.html.

## 4.2.2 Kompilace a sestavení

1. Vytvoření pracovního prostoru colcon M \_ros2 se skládá ze dvou balíčků funkcí, z nichž jeden je balíček funkcí M hal\_msgs vlastní datové struktury a druhý je balíček funkcí hlavního těla programu M \_ros2. Po instalaci základního prostředí nejprve vytvořte pracovní prostor colcon, např:

```
_{1} cd \sim/
```

mkdir -p ros2\_ws/src

2. Zkompilujte balíček funkcí Zkopírujte kód instalačního balíčku do adresáře ros2\_ws/src a v adresáři ros2\_ws spusťte následující příkaz:

<sup>1</sup> colcon build --packages-select Mhal\_msgs

Po vyčkání na dokončení kompilace předchozího příkazu zadejte:

<sup>1</sup> colcon build --packages-select M\_ros2

# 4.3 Rychlý start

### 4.3.1 Start

Otevřete příkazový řádek pod Ubuntu a zadejte:

```
cd ros2_ws
```

1

2

3

1

```
source install/setup.bash
```

ros2 run M\_ros2 ros2\_cmd\_server

## 4.3.2 Zobrazení zpětné vazby o stavu robotického ramene

Zpětná vazba o stavu robotického ramene se uvolňuje prostřednictvím tématu. Uživatelé mohou sledovat stavové údaje reMesh prostřednictvím vestavěného příkazu ros2 nebo napsat program pro jejich získání. Následující příklad ukazuje, jak pozorovat stavová data robotického ramene prostřednictvím příkazu ros2.

Otevřete příkazový řádek pod Ubuntu a zadejte:

- cd ros2\_ws
- <sup>2</sup> source install/setup.bash
- <sup>3</sup> ros2 topic echo /nonrt\_state\_data

Údaje o stavu můžete neustále sledovat v okně příkazového řádku, jak je znázorněno na obrázku níže.

. A	fr@fr-ThinkPad-E14: ~/ros2_ws	Q	ille	۵	×
1. 1.5.7.5.					
prg_state: 1					
error code: 0					
robot_mode: 1					
j1 cur pos: 7.175921440124512					
j2_cur_pos: -92.97022247314453					
j3 cur pos: 84.50509643554688					
j4_cur_pos: -98.47774505615234					
j5_cur_pos: -89.92124938964844					
j6_cur_pos: 28.24356460571289					
cart_x_cur_pos: -377.9674987792969					
cart_y_cur_pos: -151.0094757080078					
cart_z_cur_pos: 245.93238830566406					
cart_a_cur_pos: -165.01219177246094					
cart_b_cur_pos: 7.993653297424316					
cart_c_cur_pos: 69.99728393554688					
tool_num: 1					
j1_cur_tor: -4.0					
j2_cur_tor: -152.0					
j3_cur_tor: -138.0					
j4_cur_tor: -14.4					
j5_cur_tor: 0.0					
j6_cur_tor: 8.0					
prg_name: /fruser/testzwb.lua					
prg_total_line: 0					
prg_cur_line: 0					
dgt_output_h: 0					
dgt_output_l: 0					
tl_dgt_output_l: 0					
dgt_input_h: 0					
dgt_input_l: 0					
tl_dgt_input_L: 17					
rt_rx_data: 0.0					
ft_fy_data: 0.0					
rt_rz_data: 0.0					
ft_tk_data: 0.0					
ft tr data: 0.0					
ft actesature 0					200
cobot motion done: 1					
acia motion dono. A					
grup_Morton_bone. o					

# 4.3.3 Vydání příkazu

Otevřete příkazový řádek pod Ubuntu a zadejte:

```
cd ros2_ws
source install/setup.bash
```

3 rqt

Po provedení výše uvedeného příkazu se vyvolá grafické rozhraní rqt, jak je znázorněno na obrázku níže.



V rozhraní GUI zvolte pluginy->serivce->serivce caller a vyvolejte následující rozhraní, /M\_ROS\_API\_service, zadejte řetězec příkazů do výrazu rozhraní a klikněte na tlačítko volat, aby se v následujícím dialogovém okně zobrazila zpráva s odpovědí. vyberte

File Plugi	ns Pupping	Default - rqt	00
Service Ca	aller	Tempetities Help	□⊕≳@ - 0
Servic	e /FR ROS /	API service	✓ Call
Request			
Торіс		Туре	Expression
Pesnonse			
Response Field	Туре	Value	
Response Field	Туре	Value	

#### Důležité:

• Zadejte řetězcový popis pravidla:

Program interně kontroluje formát vstupního řetězce. Formát vstupního řetězce funkce musí být ve tvaru [název funkce]() a řetězec parametrů v závorkách musí být složen z písmen, číslic, čárek a minusek. Jiné znaky nebo mezery ohlásí chybu.

• Popis hodnoty zpětné vazby příkazu:

Kromě příkazu GET, který poskytne zpětnou vazbu v podobě řetězce řetězců, jsou všechny zpětné hodnoty ostatních funkcí typu int. Obecně platí, že 0 znamená, že došlo k chybě, a 1 znamená, že příkaz byl proveden správně. Pokud existují jiné hodnoty, podívejte se na kód chyby odpovídající kódu chyby definovanému v xmlrpc SDK.

# 4.3.4 Úprava parametru

Protože zjednodušené rozhraní SDK je vylepšením původního rozhraní SDK, může být zjednodušeno, protože některé parametry mají výchozí hodnoty a v procesu skutečného používání nastanou situace, kdy výchozí parametry nemohou splnit požadavky. V této chvíli můžete hodnoty příslušných výchozích parametrů upravit., a poté je načtete do uzlu.

V souboru zdrojového kódu je soubor parametrů M\_ros2\_para.yaml. Parametry v tomto souboru jsou přednastavené výchozí param etry, které se používají pro zjednodušení vstupních parametrů příkazů. Parametry můžete upravit podle svých konkrétních potřeb a poté je dynamicky upravit pomocí příkazu: ros2 param load M\_ROS\_API\_nod ~/ros2\_ws/src/M\_ros2\_para.yaml.

# 4.4 API Popis

```
/*
1
   funkce bref:store a joint space point
2
3
id
   - index bodu, začátek M om 1. uvědomte si, že tento id je závislý M om bodu, j
    \rightarrowid funkce CARTPoint
   double j1-j6 - poloha 6 os, jednotka je deg
4
5
   int JNTPoint(int id, double j1, double j2, double j3, double j4, double j5, double j6)
6
   // příklad
7
   JNTPoint(1,10,11,12,13,14,15)
8
9
10
   funkce bref:store bod kartézského prostoru
11
   id - index bodu, začátek M om 1, uvědomte si, že toto id je závislé M om the
12
     \rightarrow id_{bodu} funkce JNTPoint 
   double x,y,z,rx,ry,yz - kartézská poloha, jednotka vzdálenosti je mm, jednotka úhlu je deg
13
14
   int CARTPoint(int id, double x,y,z,rx,ry,rz)
15
   // příklad
16
   CARTPoint(1,100,110,200,0,0,0)
17
18
19
   funkce bref:get the specific id point data of joint space or cartesian space
20
   název řetězce – vstup 'JNT' nebo 'CART',JNT znamená společný prostorový bod, 'CART' znamená kartézský
21
    ↔prostorový bod
   int id - id bodu, začíná Mom l
22
23
   string GET(string name, int id)
24
   // příklad
25
   GET(JNT,1)
26
27
28
   funkce bref: Mee přepínač režimu pohonu
29
   uint8 t state - 1-open M ee drive mode, 0-close M ee drvie mode
30
31
   int DragTeachSwitch(uint8 t state)
32
   // příklad
33
   DragTeachSwitch(0)
34
35
                                                                                        (pokračování na další straně)
```

(pokračování na předchozí straně)

```
36
   funkce bref:robot servo on switch
37
   uint8 t state - 1-servo on,0-servo off
38
39
   int RobotEnable(uint8 t state)
40
   // příklad
41
   RobotEnable(1)
42
43
44
   funkce bref:přepínač provozního režimu robota
45
   uint8 t state - 1-ruční režim,0-automatický režim
46
    */
47
   int Mode(uint8 t state)
48
   // příklad
49
   \operatorname{Re}\check{z}im(1)
50
51
   /*
52
   funkce bref:nastavit rychlost robota při aktuální
                                                             režim
53
   oneraci
   float vel - procento rychlosti, Mom 1 až 100
54
55
   int SetSpeed(float vel)
56
   // příklad
57
   SetSpeed(10)
58
59
   /*
60
   funkce bref:set and load specific index tool coordinate
61
   int id - index souřadnice nástroje, M om 1 až 15
62
   float x,y,z,rx,ry,rz - transformace souřadnic nástroje
63
64
   int SetToolCoord(int id, float x, float y, float z, float rx, float ry, float rz)
65
   // příklad
66
   SetToolCoord(1,0,0,0,0,0,0,0,0)
67
68
69
   funkce bref:set seznam souřadnic nástroje
70
   int id - index seznamu souřadnic nástroje, Mom 1 až 15
71
   float x,y,z,rx,ry,rz - transformace souřadnic nástroje
72
   */
73
   int SetToolList(int id, float x,float y, float z,float rx,float ry,float rz);
74
   // příklad
75
   SetToolList(1,0,0,0,0,0,0,0,0)
76
77
   /*
78
   funkce bref:set and load specific index externí souřadnice nástroje
79
   int id - index souřadnice vnějšího nástroje, M om 1 až 15
80
   float x,y,z,rx,ry,rz - transformace vnější souřadnice nástroje
81
82
   int SetExToolCoord(int id, float x,float y, float z,float rx,float ry,float rz);
83
   // příklad
84
   SetExToolCoord(1,0,0,0,0,0,0,0,0)
85
86
```

87

```
(pokračování na předchozí straně)
```

```
funkce bref:set seznam souřadnic externího nástroje
88
    int id - index souřadnice vnějšího nástroje, Mom 1 až 15
89
    float x,y,z,rx,ry,rz - transformace vnější souřadnice nástroje
90
    */
91
    int SetExToolList(int id, float x, float y, float z, float rx, float ry, float rz);
92
    // přiklad
93
    SetExToolList(1,0,0,0,0,0,0,0,0)
94
95
    /*
96
    funkce bref:set souřadnice objektu
97
    int id - index souřadnice objektu, Mom 1 až 15
98
    float x,y,z,rx,ry,rz - transformace souřadnic objektu
99
100
    int SetWObjCoord(int id, float x, float y, float z, float rx, float ry, float rz);
101
    // priklad
102
    SetWObjCoord(1,0,0,0,0,0,0,0,0)
103
104
105
    funkce bref:set seznam souřadnic objektu
106
    int id - index souřadnice objektu, Mom 1 až 15
107
    float x,y,z,rx,ry,rz - transformace souřadnic objektu
108
109
    int SetWObjList(int id, float x,float y, float z,float rx,float ry,float rz);
110
    // příklad
111
    SetWObjList(1,0,0,0,0,0,0,0,0)
112
113
114
    funkce bref:set TCP load weight
115
    hmotnost plováku - hmotnost nákladu, jednotka je kg
116
117
    int SetLoadWeight(float weight);
118
    // příklad
119
    SetLoadWeight(3.5)
120
121
122
    funkce bref:set hmotnost těžiště nákladu
123
    float x,y,z - umístění těžiště, uint je mm
124
125
    int SetLoadCoord(float x,float y,float z);
126
    // příklad
127
    SetLoadCoord(10,20,30)
128
129
130
    funkce bref:set robot install direction
131
    uint8 t install - 0-podlaha, 1-stěna, 2-strop
132
133
    int SetRobotInstallPos(uint8_t install);
134
    // příklad
135
    SetRobotInstallPos(0)
136
137
138
    funkce bref:set robot installation dirction v případě M ee install
139
```

(pokračování na další straně)

140

141

142

143

144

145 146

147

148

149 150

151

152

153 154

155

156

157

158

159

160

161 162 163

164

165

166

167

168

169 170 171

172

173

174

175

176

177 178 179

180

181

182 183 184

185

186

187

188

```
dvojitý úhel - úhel ponoru
double zangle - úhel natočení
*/
int SetRobotInstallAngle(double yangle,double zangle);
// příklad
SetRobotInstallAngle(90,0)
/*
funkce bref:set axies collision levels
float level1-level6 - úroveň kolize každé osy, Mom 1 až 10
int SetAnticollision(float level1, float level2, float level3, float level4, float
, ouroveň5, folát úroveň6);
// příklad
SetAnticollision(1,1,1,1,1,1,1)
/*
funkce bref:set strategy after collision
int strategie - 0- zastavit pohyb a vyhodit chybu, 1-pokračovat v běhu
*/
int SetCollisionStrategy(int strategy);
// příklad
SetCollisionStrategy(1)
funkce bref:nastavit kladnou mez každé osy
float limit1-limit6 - hodnota limitu každé osy
*/
int SetLimitPositive(float limit1, float limit2, float limit3, float
                                                                        limit4, float
→limit5, float limit6);
// příklad
SetLimitPositve(100,90,90,90,90,90)
funkce bref:nastavit zápornou mez každé osy
float limit1-limit6 - hodnota limitu každé osy
*/
int SetLimitNegative(float limit1, float limit2, float limit3, float
                                                                        limit4, float
→limit5, float limit6);
// příklad
SetLimitNegative(-100,-90,-90,-90,-90)
funkce bref:error state clear
*/
int ResetAllError();
funkce bref: joint M iction compensation switch
uint8 t state - 0-off, 1-on
*/
int MictionCompensationOnOff(uint8_t state);
```

(pokračování na další straně)

(pokračování na předchozí straně)

```
(pokračování na předchozí straně)
    // příklad
189
    MictionCompensationOnOff(1)
190
191
192
    funkce bref:set koeficient každého spoje v případě pokládky podlahy
193
194
    float coeff1-coeff6 - koeficient každého spoje, Mom 0 až 1
195
    int SetMictionValue_level(float coeff1,float coeff1,float coeff3,float coeff4,float,
196
      coeff5, float coeff6);
    // příklad
197
    SetMictionValue_level(1,1,1,1,1,1,1)
198
199
    /*
200
    funkce bref:nastavit koeficient každého spoje v případě instalace na stěnu
201
    float coeff1-coeff6 - koeficient každého spoje, M om 0 až 1
202
203
    int SetMictionValue_wall(float coeff1,float coeff1,float coeff3,float coeff4,float_
204
      >coeff5, float coeff6);
    // příklad
205
    SetMictionValue_wall(0.5,0.5,0.5,0.5,0.5,0.5)
206
207
208
    funkce bref:nastavit koeficient každého spoje v případě stropní instalace
209
    float coeff1-coeff6 - koeficient každého spoje, Mom 0 až 1
210
211
    int SetMictionValue_ceiling(float coeff1,float coeff1,float coeff3,float coeff4,float_
212
      coeff5, float coeff6);
    // příklad
213
    SetMictionValue_ceiling(0.5,0.5,0.5,0.5,0.5,0.5,0.5)
214
215
216
    funkce bref: active gripper
217
    int index - index chapadla
218
    uint8 t act - 0-reset, 1-aktivní
219
    */
220
    int ActGripper(int index,uint8_t act);
221
    // příklad
222
    ActGripper(1,1)
223
224
225
    funkce bref:řízení pohybu chapadla
226
    int index - index chapadla
227
    int pos - trvalý údaj o poloze chapadla, Mom 0 až 100
228
229
    int MoveGripper(int index,int pos);
230
    // příklad
231
    MoveGripper(1,10)
232
233
    /*
234
    funkce bref:nastavit digitální výstup řídicí jednotky
235
    int id - index IO, Mom 0 až 15
236
    uint t status - 0-off, 1-on
237
```

(pokračování na další straně)

(pokračování na předchozí straně)

```
*/
238
    int SetDO(int id,uint8_t status);
239
    // příklad
240
    SetDO(1,1)
241
242
    /*
243
    funkce bref:set digitální výstup nástroje
244
245
    int id - index IO, Mom 0 až 1
246
    uint t status - 0-off, 1-on
247
248
    int SetToolDO(int id,uint8_t status);
249
    // příklad
250
    SetToolDO(0,1)
251
252
    /*
253
    funkce bref:set analog output of control box int id
254
    - index IO, M om 0 až 1
255
    float vlaue - proud napěťové perzistence, Mom 0 až 100
256
257
    int SetAO(int id,float value);
258
    // příklad
259
    SetAO(1,100)
260
261
    /*
262
    funkce bref:set analogový výstup nástroje
263
    int id - index IO, Mom 0 až 0
264
    float vlaue - proud napěťové perzistence, Mom 0 až 100
265
    */
266
267
    int SetToolAO(int id,float value);
268
    // příklad
269
    SetToolAO(0,100)
270
271
    /*
    funkce bref:JOG
    uint8 t ref - 0-kloubový souřadnicový běh, 2-základní souřadnicový běh, 4-nástrojový souřadnicový běh,
272
8-
    ↔ souřadnicový běh objektu
    uint8 t nb - losal(osa x),2-osa2(osa y),3-osa3(osa z),4-osa4(rx),5-osa5(ry),6-osy5(ry),6
273
    \rightarrow os6(rz)
    uint8 t dir - 0-negativní směr, 1-pozitivní směr
274
    float vel - trvalá rychlost, M om 0 až 100
275
276
    int StartJOG(uint8 t ref, uin8_t nb, uint8 t dir, float vel);
277
    // příklad
278
    StartJOG(1,1,1,10)
279
280
281
    funkce bref:JOG stop
282
    uint8 t ref - 0-kloubový souřadnicový jog stop, 2-základní souřadnicový jog stop, 4-nástrojový
283
souřadnicový jog stop
       jog stop, 8-objektový souřadnicový jog stop
    */
284
```

- int StopJOG(uint8\_t ref);
  // příklad 285
- 286

(pokračování na další straně)

```
(pokračování na předchozí straně)
    StopJOG(1)
287
288
    /*
289
    funkce bref: JOG okamžitě zastavit
290
291
292
    int ImmStopJOG();
293
    /*
294
    funkce bref:pohyb z bodu do bodu v kloubním
295
    prostoru
    string point name - název předřazeného bodu, například JNTI znamená první bod kloubu
296
    ↔ prestorovaný bod, CART znamená první bod pro cartieovský prestorovaný bod
    float vel - trvalá rychlost, Mom 0 až 100
297
298
    int MoveJ(string point_name, float vel);
299
    // příklad
300
    MoveJ(JNT1,10)
301
302
303
    funkce bref:lineární pohyb v kartézském prostoru
304
    string point name – název předřazeného bodu, například JNTI znamená první bod kloubu
305
    ↔ prestorovaný bod, CART znamená první bod pro cartieovský prestorovaný bod
    float vel - trvalá rychlost, Mom 0 až 100
306
    */
307
    int MoveL(string point_name,float vel);
308
    // příklad
309
    MoveL(CART1,10)
310
311
312
    funkce bref: arc motion v kartézském prostoru
313
    string point1 name point2 name - název předřazeného bodu, například JNT1 znamená první bod
314
    ↔ společného předpjatého bodu, CART znamená první bod kartézského předpjatého bodu, be
    ↔ uvědomit si, že dva body musí být stejného typu, což znamená, že uživatel musí zadat dva INT.
    \leftrightarrowbody nebo dva body CART
    float vel - trvalá rychlost, M om 0 až 100
315
316
    int MoveC(string point1_name, string point2_name, float vel);
317
    // příklad
318
    MoveC(JNT1, JNT2, 10)
319
320
321
    funkce bref: joint space spline motion start
322
323
    int SplineStart();
324
325
    /*
326
    funkce bref:Spline pohyb v kloubním prostoru, podporován pouze bod JNT, dojde k chybě
327
     ↔vyhozeno, pokuď je na vstupu bod CART
    string point name – název předřazeného bodu, například JNTI znamená první bod kloubu
328
    ↔obnovený bod
    float vel - trvalá rychlost, Mom 0 až 100
329
330
    int SplinePTP(string point_name, float vel);
331
```

(pokračování na další straně)
(pokračování na předchozí straně)

```
// příklad
332
333
    SplinePTP(JNT2,10)
334
335
    funkce bref: joint space spline motion end
336
    */
337
    int SplineEnd();
338
339
    /*
340
    funkce bref:cartesian space spline motion start
341
    uint8 t ctlpoint - 0-trajektorie přes řídicí bod, 1-trajektorie nedosáhne
342
    •→kontrolní bod
343
    int NewSplineStart(uint8_t ctlpoint);
344
    // příklaď
345
    NewSplineStrart(1)
346
347
    /*
348
    funkce bref:Spline motion v kartézském prostoru, podporován pouze bod CART, dojde k chybě
349
     ↔bude vyhozen, pokud je na vstupu bod JNT
    string point name - název předvyplněného bodu, například CARTI znamená první bod.
350
    ↔kartézský předsazený bod
    float vel - trvalá rychlost, Mom 0 až 100
351
    int lastflag - 0-není poslední bod, 1-poslední bod
352
353
    int NewSplinePoint(string point_name, float vel, int lastflag);
354
    // příklaď
355
    NewSplinePoint(JNT2,20,0)
356
357
358
    funkce bref:cartesian space spline motion end
359
360
    int NewSplineEnd();
361
362
363
    funkce bref:stop robot motion
364
365
    int StopMotion();
366
367
    /*
368
    funkce bref:points shift start
369
    int flag - 0-posun na souřadnici základny/objektu, 2-posun na souřadnici nástroje
370
    double x,y,z,rx,ry,rz - transformace posunu
371
372
    int PointsOffsetEnable(int flag,double x,double y,double z,double rx,double ry,double
373
    ,,
// příklad
374
    PointsOffsetEnable(1,10,10,10,0,0,0)
375
376
377
    funkce bref:points shift end
378
    */
379
```