

Software version : V3.2.2  
Document version : V3.2.2  
Original instructions(Korean)

# Programming manual(V3.2.2)



M0609 | M0617 | M1013 | M1509 | H2017 |  
H2515 | A0509 | A0509S | A0912 | A0912S

**DOOSAN**

© 2024 Doosan Robotics Inc.

---

## Table of Contents

<b>1</b>	<b>머리말 .....</b>	<b>15</b>
1.1	저작권 .....	15
<b>2</b>	<b>DRL Basic Syntax .....</b>	<b>16</b>
2.1	기본 문법 .....	16
2.1.1	들여쓰기 .....	16
2.1.2	주석 .....	17
2.2	변수 .....	17
2.2.1	변수명 .....	17
2.2.2	수치 .....	18
2.2.3	문자 .....	18
2.2.4	list .....	20
2.2.5	tuple .....	20
2.2.6	dictionary .....	21
2.3	함수 .....	21
2.3.1	함수 문법 .....	21
2.3.2	스코핑 블록 .....	22
2.3.3	인수 모드 .....	23
2.4	제어문 .....	24
2.4.1	pass .....	24
2.4.2	if .....	24
2.4.3	while .....	25
2.4.4	for .....	26
2.4.5	break .....	26
2.4.6	continue .....	27
2.4.7	반복문의 else .....	27
<b>3</b>	<b>모션 관련 명령어.....</b>	<b>29</b>
3.1	위치 생성 .....	29
3.1.1	posj(J1=0, J2=0, J3=0, J4=0, J5=0, J6=0) .....	29
3.1.2	posx(x=0, y=0, z=0, A=0, B=0, C=0, ori_type=None, sol=None, turn=None) .....	30

---

3.1.3	trans(pos, delta, ref, ref_out, ori_type_out) .....	32
3.1.4	posb(seg_type, posx1, posx2=None, radius=0) .....	35
3.1.5	fkin(pos, ref, ori_type) .....	37
3.1.6	ikin(pos, sol_space, ref, ref_pos_opt, iter_threshold) .....	39
3.1.7	addto(pos, add_val=None) .....	41
3.1.8	ikin_norm(pos, sol_space, ref, ref_pos_opt) .....	43
3.2	모션 설정 .....	45
3.2.1	set_velj(vel) .....	45
3.2.2	set_accj(acc) .....	47
3.2.3	set_velx(vel1, vel2) .....	48
3.2.4	set_accx(acc1, acc2) .....	50
3.2.5	set_tcp(name) .....	52
3.2.6	set_ref_coord(coord) .....	53
3.2.7	set_auto_acceleration_mode(mode, ratio) .....	55
3.3	동기 모션 .....	57
3.3.1	movej() .....	57
3.3.2	movel() .....	62
3.3.3	movejx() .....	67
3.3.4	movec() .....	71
3.3.5	movesj() .....	77
3.3.6	movesx() .....	81
3.3.7	moveb() .....	85
3.3.8	move_spiral() .....	90
3.3.9	move_periodic() .....	94
3.3.10	move_home() .....	99
3.4	비동기 모션 .....	100
3.4.1	amovej() .....	100
3.4.2	amovel() .....	103
3.4.3	amovejx() .....	106
3.4.4	amovec() .....	109
3.4.5	amovesj() .....	113
3.4.6	amovesx() .....	116
3.4.7	amoveb() .....	119
3.4.8	amove_spiral() .....	124
3.4.9	amove_periodic() .....	128

---

---

<b>3.5</b>	<b>추가 기능 .....</b>	<b>132</b>
3.5.1	mwait(time=0).....	132
3.5.2	begin_blend(radius=0) .....	133
3.5.3	end_blend().....	135
3.5.4	check_motion() .....	136
3.5.5	stop(st_mode) .....	138
3.5.6	change_operation_speed(speed) .....	140
3.5.7	wait_manual_guide() .....	142
3.5.8	wait_nudge() .....	144
3.5.9	enable_alter_motion(n,mode,ref,limit_dPOS,limit_dPOS_per) .....	146
3.5.10	alter_motion(pos).....	148
3.5.11	disable_alter_motion().....	150
3.5.12	check_robot_mastering() .....	152
3.5.13	motion_pause() .....	153
3.5.14	motion_resume() .....	155
<b>3.6</b>	<b>서보 모션 .....</b>	<b>156</b>
3.6.1	servoj() .....	156
3.6.2	servol() .....	159
3.6.3	speedj().....	161
3.6.4	speedl().....	163

---

<b>4</b>	<b>제어 보조 명령어.....</b>	<b>167</b>
<b>4.1</b>	<b>로봇 현재값 .....</b>	<b>167</b>
4.1.1	get_current_posj() .....	167
4.1.2	get_current_velj().....	167
4.1.3	get_current_posx(ref, ori_type) .....	168
4.1.4	get_current_tool_flange_posx(ref, ori_type) .....	170
4.1.5	get_current_velx(ref).....	172
4.1.6	get_current_rotm(ref).....	173
4.1.7	get_joint_torque() .....	174
4.1.8	get_external_torque() .....	174
4.1.9	get_tool_force(ref) .....	175
<b>4.2</b>	<b>로봇 목표값 .....</b>	<b>176</b>
4.2.1	get_desired_posj() .....	176
4.2.2	get_desired_velj() .....	177

---

4.2.3	get_desired_posx(ref, ori_type) .....	178
4.2.4	get_desired_velx(ref) .....	179
4.3	제어 상태값 .....	180
4.3.1	get_control_mode() .....	180
4.3.2	get_control_space() .....	181
4.3.3	get_current_solution_space() .....	181
4.3.4	get_solution_space(pos) .....	182
4.3.5	get_orientation_error(xd, xc, axis) .....	183

## 5 기타 설정 명령어.....185

5.1	툴/작업물 설정 .....	185
5.1.1	get_workpiece_weight() .....	185
5.1.2	reset_workpiece_weight().....	186
5.1.3	set_workpiece_weight(weight=0.0, cog=[0.0,0.0,0.0], cog_ref=DR_CUR_TCP, add_up=DR_REPLACE, start_time=None, transition_time=None).....	186
5.1.4	set_tool_shape(name) .....	189
5.1.5	set_tool(name, start_time, transition_time) .....	190
5.2	제어 모드 설정.....	191
5.2.1	set_singularity_handling(mode) .....	191
5.2.2	set_singular_handling_force(mode).....	193
5.2.3	set_palletizing_mode(mode) .....	195
5.2.4	set_motion_end(mode) .....	196

## 6 힘/순응 제어 및 기타 사용자 편의 기능 .....199

6.1	힘/순응 제어 .....	199
6.1.1	release_compliance_ctrl() .....	199
6.1.2	task_compliance_ctrl(stx, time).....	200
6.1.3	set_stiffnessx(stx, time) .....	201
6.1.4	set_desired_force(fd, dir, time, mod) .....	203
6.1.5	release_force(time=0) .....	206
6.1.6	get_force_control_state().....	207
6.1.7	set_damping_factor(damping_factor, time) .....	209
6.1.8	set_force_factor(force_factor, time) .....	211
6.2	사용자 편의 기능.....	213
6.2.1	parallel_axis(x1, x2, x3, axis, ref) .....	213

---

6.2.2	parallel_axis(vect, axis, ref).....	214
6.2.3	align_axis(x1, x2, x3, pos, axis, ref) .....	216
6.2.4	align_axis(vect, pos, axis, ref) .....	218
6.2.5	is_done_bolt_tightening(m=0, timeout=0, axis=None).....	219
6.2.6	calc_coord(x1, x2, x3, x4, ref, mod, ori_type_out) .....	221
6.2.7	set_user_cart_coord(pos, ref) .....	223
6.2.8	set_user_cart_coord(x1, x2, x3, pos, ref) .....	224
6.2.9	set_user_cart_coord(u1, v1, pos, ref) .....	226
6.2.10	overwrite_user_cart_coord(id, pos, ref, apply_mod) .....	227
6.2.11	get_user_cart_coord(id, ori_type).....	229
6.2.12	check_position_condition(axis, min, max, ref, mod, pos) .....	230
6.2.13	check_force_condition(axis, min, max, ref).....	233
6.2.14	check_orientation_condition(axis, min, max, ref, mod) .....	234
6.2.15	check_orientation_condition(axis, min, max, ref, mod, pos) .....	237
6.2.16	coord_transform(pose_in, ref_in, ref_out, ori_type_out) .....	240
6.2.17	get_pattern_point(pos1, pos2, pos3, pos4, index, pattern, row, column, stack, stack_offset, point_offset, ori_type_out=None) .....	242

## 7 시스템 명령어 ..... 246

7.1	IO 관련 .....	246
7.1.1	set_digital_output(index, val=None) .....	246
7.1.2	set_digital_outputs(bit_list) .....	247
7.1.3	set_digital_outputs(bit_start, bit_end, val) .....	248
7.1.4	set_digital_output(index, val=None, time=None, val2=None).....	250
7.1.5	get_digital_input(index) .....	251
7.1.6	get_digital_inputs(bit_list).....	252
7.1.7	get_digital_inputs(bit_start, bit_end) .....	254
7.1.8	wait_digital_input(index, val, timeout=None).....	255
7.1.9	set_tool_digital_output(index, val=None) .....	256
7.1.10	set_tool_digital_outputs(bit_list) .....	258
7.1.11	set_tool_digital_outputs(bit_start, bit_end, val).....	259
7.1.12	set_tool_digital_output(index, val=None, time=None, val2=None) .....	260
7.1.13	get_tool_digital_input(index) .....	262
7.1.14	get_tool_digital_inputs(bit_list) .....	263
7.1.15	get_tool_digital_inputs(bit_start, bit_end).....	264

---

7.1.16	wait_tool_digital_input(index, val, timeout=None) .....	266
7.1.17	set_mode_analog_output(ch, mod ) .....	267
7.1.18	set_mode_analog_input(ch, mod ) .....	268
7.1.19	set_analog_output(ch, val).....	269
7.1.20	get_analog_input(ch).....	271
7.1.21	set_output(port_type, index, val=None, time=None, val2=None).....	272
7.1.22	get_input(port_type, index) .....	274
7.1.23	wait_input(port_type, index, val, timeout=None, condition=None) .....	276
7.1.24	wait_analog_input(ch, condition, val, timeout=None) .....	278
7.1.25	wait_tool_analog_input(ch, condition, val, timeout=None).....	279
7.2	<b>TP 연동</b> .....	281
7.2.1	tp_popup(message, pm_type=DR_PM_MESSAGE, button_type=0) .....	281
7.2.2	tp_log(message) .....	282
7.2.3	tp_get_user_input(message, input_type) .....	283
7.3	<b>Thread</b> .....	285
7.3.1	thread_run(th_func_name, loop=False).....	285
7.3.2	thread_stop(th_id) .....	287
7.3.3	thread_pause(th_id) .....	288
7.3.4	thread_resume(th_id) .....	289
7.3.5	thread_state(th_id) .....	290
7.3.6	통합 예제 - Thread .....	291
7.4	<b>기타</b> .....	293
7.4.1	wait(time) .....	293
7.4.2	exit().....	294
7.4.3	sub_program_run(name) .....	294
7.4.4	drl_report_line(option) .....	297
7.4.5	set_fm(key, value) .....	298
7.4.6	get_robot_model().....	299
7.4.7	get_robot_serial_num() .....	299
7.4.8	check_robot_jts() .....	300
7.4.9	check_robot_fts() .....	300
7.4.10	start_timer() .....	301
7.4.11	end_timer() .....	302
7.4.12	message_to_dp(event_name, strdata) .....	302
7.4.13	send_load_module(package, sub_type, active) .....	303

---

---

7.4.14	send_unload_module(unique_id).....	305
--------	------------------------------------	-----

## 8 수학 함수 ..... 307

8.1	기본 함수 .....	307
8.1.1	ceil(x) .....	307
8.1.2	floor(x) .....	307
8.1.3	pow(x, y).....	308
8.1.4	sqrt(x).....	309
8.1.5	log(x, b) .....	309
8.1.6	d2r(x) .....	310
8.1.7	r2d(x) .....	311
8.1.8	random().....	311
8.2	삼각함수 .....	312
8.2.1	sin(x) .....	312
8.2.2	cos(x).....	313
8.2.3	tan(x).....	313
8.2.4	asin(x) .....	314
8.2.5	acos(x).....	315
8.2.6	atan(x).....	315
8.2.7	atan2(y, x).....	316
8.3	선형대수 .....	317
8.3.1	norm(x).....	317
8.3.2	rotx(angle) .....	317
8.3.3	rot(y, angle) .....	318
8.3.4	rot(z, angle) .....	319
8.3.5	rotm2eul(rotm) .....	319
8.3.6	rotm2rotvec(rotm) .....	320
8.3.7	eul2rotm([alpha,beta,gamma]) .....	321
8.3.8	eul2rotvec([alpha,beta,gamma]) .....	322
8.3.9	eul2rpy([alpha,beta,gamma]).....	323
8.3.10	rpy2eul([yaw,pitch,roll]) .....	324
8.3.11	rotvec2eul([rx,ry,rz]) .....	324
8.3.12	rotvec2rotm([rx,ry,rz]) .....	325
8.3.13	htrans(posx1,posx2,ori_type_out) .....	326
8.3.14	get_intermediate_pose(posx1,posx2,alpha,ori_type_out) .....	328

---

8.3.15	get_distance(posx1, posx2) .....	329
8.3.16	get_normal(x1, x2, x3) .....	330
8.3.17	add_pose(posx1, posx2, ori_type_out) .....	331
8.3.18	subtract_pose(posx1, posx2, ori_type_out).....	333
8.3.19	inverse_pose(posx1, ori_type_out) .....	334
8.3.20	dot_pose(posx1, posx2) .....	335
8.3.21	cross_pose(posx1, posx2).....	336
8.3.22	unit_pose(posx1) .....	337

## 9 외부 통신 명령어.....**339**

9.1	Serial .....	<b>339</b>
9.1.1	serial_open(port=None, baudrate=115200, bytesize=DR_EIGHTBITS, parity=DR_PARITY_NONE, stopbits=DR_STOPBITS_ONE) .....	339
9.1.2	serial_close(ser) .....	340
9.1.3	serial_state(ser) .....	341
9.1.4	serial_set_inter_byte_timeout(ser, timeout=None).....	342
9.1.5	serial_write(ser, tx_data) .....	343
9.1.6	serial_read(ser, length=-1, timeout=-1) .....	344
9.1.7	serial_get_count() .....	346
9.1.8	serial_get_info(id) .....	347
9.1.9	통합 예제 - Serial.....	347
9.1.10	플랜지 I/O .....	349
9.2	Tcp/Client .....	<b>356</b>
9.2.1	client_socket_open(ip, port).....	356
9.2.2	client_socket_close(sock).....	357
9.2.3	client_socket_state(sock).....	358
9.2.4	client_socket_write(sock, tx_data) .....	359
9.2.5	client_socket_read(sock, length=-1, timeout=-1) .....	360
9.2.6	통합 예제 (Tcp/Client) .....	362
9.3	Tcp/Server .....	<b>364</b>
9.3.1	server_socket_open(port) .....	364
9.3.2	server_socket_close(sock).....	365
9.3.3	server_socket_state(sock).....	366
9.3.4	server_socket_write(sock, tx_data) .....	367
9.3.5	server_socket_read(sock, length=-1, timeout=-1) .....	368

---

9.3.6	통합 예제 - Tcp/Server .....	369
9.4	Modbus .....	372
9.4.1	add_modbus_signal(ip, port, name, reg_type, index, value=0, slaveid=255) .....	372
9.4.2	add_modbus_rtu_signal(slaveid=1, port=None, baudrate=115200, bytesize=DR_EIGHTBITS, parity=DR_PARITY_NONE, stopbits=DR_STOPBITS_ONE, name, reg_type, index, value=0) .....	375
9.4.3	add_modbus_signal_multi(ip, port, slaveid=255, name=None, reg_type=DR_HOLDING_REGISTER, start_address=0, cnt=1) .....	377
9.4.4	add_modbus_rtu_signal_multi(slaveid=1, port=None, baudrate=115200, bytesize=DR_EIGHTBITS, parity=DR_PARITY_NONE, stopbits=DR_STOPBITS_ONE, name=None, reg_type=DR_HOLDING_REGISTER, start_address=0, cnt=1) .....	379
9.4.5	del_modbus_signal(name) .....	381
9.4.6	del_modbus_signal_multi(name) .....	382
9.4.7	set_modbus_output(iobus, value) .....	383
9.4.8	set_modbus_outputs(iobus_list, val_list) .....	385
9.4.9	set_modbus_output_multi(iobus, val_list) .....	386
9.4.10	get_modbus_input(iobus) .....	387
9.4.11	get_modbus_inputs(iobus_list) .....	389
9.4.12	get_modbus_inputs_list(iobus_list) .....	390
9.4.13	get_modbus_input_multi (iobus) .....	391
9.4.14	wait_modbus_input(iobus, val, timeout=None) .....	392
9.4.15	set_modbus_slave(address, val) .....	394
9.4.16	get_modbus_slave(address) .....	395
9.4.17	modbus_crc16(data) .....	396
9.4.18	modbus_send_make(send_data) .....	397
9.4.19	modbus_recv_check(recv_data) .....	398
9.4.20	modbus_unsigned_to_signed(unsigned_data) .....	399
9.5	Industrial Ethernet (EtherNet/IP,PROFINET) .....	400
9.5.1	set_output_register_bit(address, val) .....	400
9.5.2	set_output_register_int(address, val) .....	401
9.5.3	set_output_register_float(address, val) .....	402
9.5.4	get_output_register_bit(address) .....	403
9.5.5	get_output_register_int(address) .....	404
9.5.6	get_output_register_float(address) .....	405
9.5.7	get_input_register_bit(address) .....	406
9.5.8	get_input_register_int(address) .....	407
9.5.9	get_input_register_float(address) .....	408

---

---

<b>9.6</b>	<b>FOCAS.....</b>	<b>409</b>
9.6.1	focas_connect(ip, port, timeout).....	409
9.6.2	focas_disconnect(handle).....	410
9.6.3	focas_pmc_read_bit(handle, addr_type, start_num, bit_offset) .....	411
9.6.4	focas_pmc_read_char(handle, addr_type, start_num, read_count) .....	413
9.6.5	focas_pmc_read_word(handle, addr_type, start_num, read_count) .....	415
9.6.6	focas_pmc_read_long(handle, addr_type, start_num, read_count) .....	417
9.6.7	focas_pmc_read_float(handle, addr_type, start_num, read_count).....	419
9.6.8	focas_pmc_read_double(handle, addr_type, start_num, read_count) .....	421
9.6.9	focas_pmc_write_bit(handle, addr_type, start_num, bit_offset, write_data) .....	423
9.6.10	focas_pmc_write_char(handle, addr_type, start_num, write_data, write_count) .....	425
9.6.11	focas_pmc_write_word(handle, addr_type, start_num, write_data, write_count) .....	427
9.6.12	focas_pmc_write_long(handle, addr_type, start_num, write_data, write_count) .....	429
9.6.13	focas_pmc_write_float(handle, addr_type, start_num, write_data, write_count).....	431
9.6.14	focas_pmc_write_double(handle, addr_type, start_num, write_data, write_count) .....	433
9.6.15	focas_get_error_str(handle, errorCode) .....	435
<b>10</b>	<b>외부 비전 명령어.....</b>	<b>440</b>
10.1	개요 .....	440
10.2	<b>2D Vision - COGNEX / SICK / VISOR.....</b>	<b>440</b>
10.2.1	vs_set_info(type).....	441
10.2.2	vs_connect(ip_addr, port_num=9999) .....	442
10.2.3	vs_disconnect() .....	443
10.2.4	vs_get_job() .....	443
10.2.5	vs_set_job(job_name) .....	444
10.2.6	cognex_set_integer (job_name, integer_number) .....	445
10.2.7	visor_job_change(index) .....	445
10.2.8	vs_trigger() .....	446
10.2.9	vs_set_init_pos(vision_posx_init, robot_posx_init, vs_pos=1).....	447
10.2.10	vs_get_offset_pos(vision_posx_meas, vs_pos=1) .....	449
10.2.11	통합예제 - DR_VS_COGNEX, DR_VS_SICK.....	450
10.2.12	vs_request(cmd).....	451
10.2.13	vs_result().....	451
10.2.14	통합예제 - DR_VS_CUSTOM .....	452
10.3	Pickit 3D .....	454

---

10.3.1	pickit_connect(ip).....	455
10.3.2	pickit_disconnect() .....	455
10.3.3	pickit_change_configuration(setup_id, product_id)) .....	456
10.3.4	pickit_request_calibration() .....	456
10.3.5	pickit_detection(offset_z) .....	457
10.3.6	pickit_next_object(offset_z) .....	459
10.3.7	pickit_save_snapshot() .....	460
10.3.8	통합 예제 - Pickit 3D .....	461

## 11 두산비전(SVM) 명령어 .....463

11.1	카메라 연결 .....	464
11.1.1	svm_connect(ip="192.168.137.5", port=20).....	464
11.1.2	svm_disconnect() .....	465
11.2	이미지 품질 체크.....	465
11.2.1	svm_set_led_brightness(value) .....	465
11.2.2	svm_get_led_brightness() .....	466
11.2.3	svm_set_camera_exp_val(value) .....	466
11.2.4	svm_set_camera_gain_val(value) .....	467
11.2.5	svm_set_camera_load(job_id) .....	468
11.3	비전 작업 관리.....	468
11.3.1	svm_set_job(job_id) .....	468
11.4	카메라 보정 .....	469
11.4.1	svm_get_robot_pose(job_id) .....	469
11.5	사물 인식 및 감지.....	470
11.5.1	svm_get_vision_info(job_id).....	470
11.5.2	svm_get_variable(tool_id, var_type).....	471
11.5.3	svm_detect_landmark(job_id) .....	472
11.5.4	svm_get_marker_offset_pose(cpos, offset, euler_mode) .....	473
11.5.5	svm_get_detect_barcode() .....	474
11.5.6	svm_get_barcode_db_data(index) .....	475
11.5.7	svm_compare_barcode_db_data(dbdata) .....	475
11.6	로봇 태스크 .....	476
11.6.1	svm_set_init_pos_data(Id_list, Pos_list).....	476
11.6.2	svm_get_offset_pos(posx_robot_init, job_id, tool_id) .....	477

---

11.7	기타 SVM 명령어.....	479
11.7.1	svm_set_tp_popup(svm_flag).....	479
11.8	통합예제 - SVM.....	479
11.8.1	예제 .....	479
<b>12</b>	<b>Application 명령어.....</b>	<b>481</b>
12.1	External Encoder 설정 명령어 .....	481
12.1.1	set_extenc_polarity(channel, polarity_A, polarity_B, polarity_Z, polarity_S) .....	481
12.1.2	set_extenc_mode(channel, mode_AB, pulse_AZ, mode_Z, mode_S, inverse_cnt) .....	482
12.1.3	get_extenc_count(channel) .....	484
12.1.4	clear_extenc_count(channel) .....	485
12.2	Conveyor Tracking.....	486
12.2.1	set_conveyor(name) .....	486
12.2.2	set_conveyor_ex(name="", conv_type=0, encoder_channel=1, triggering_mute_time=0.0, count_per_dist=5000, conv_coord=posx(0,0,0,0,0,0), ref=DR_BASE, conv_speed=100.0, speed_filter_size=500, min_dist=0.0, max_dist=1000.0 ...).....	487
12.2.3	get_conveyor_obj(conv_id, timeout=None, container_type=DR_FIFO, obj_offset_coord=None).....	491
12.2.4	tracking_conveyor(conv_id, time=0.3) .....	495
12.2.5	untracking_conveyor(conv_id, time=0.3) .....	497
12.3	Welding.....	498
12.3.1	app_weld_enable_digital() .....	498
12.3.2	app_weld_disable_digital() .....	500
12.3.3	app_weld_set_interface_eip_r2m_process(welding_start=[0,0,0,0,0,0,0,0], robot_ready=[0,0,0,0,0,0,0,0], error_reset=[0,0,0,0,0,0,0,0]) .....	502
12.3.4	app_weld_set_interface_eip_r2m_mode(welding_mode=[0,0,0,0,0,0,0,0], s_2t=[0,0,0,0,0,0,0,0], pulse_mode=[0,0,0,0,0,0,0,0],wm_opt1=[0,0,0,0,0,0,0,0]) .....	507
12.3.5	app_weld_set_interface_eip_r2m_test(gas_test=[0,0,0,0,0,0,0,0], inchinc_plus=[0,0,0,0,0,0,0,0], inchinc_minus=[0,0,0,0,0,0,0,0], blow_out_torch=[0,0,0,0,0,0,0,0], simulation=[0,0,0,0,0,0,0,0], ts_opt1=[0,0,0,0,0,0,0,0], ...).....	511
12.3.6	app_weld_set_interface_eip_r2m_condition(job_num=[0,0,0,0,0,0,0,0], synergic_id=[0,0,0,0,0,0,0,0], r_wire_feed_speed=[0,0,0,0,0,0,0,0], voltage_corret=[0,0,0,0,0,0,0,0], dynamic_correct=[0,0,0,0,0,0,0,0]) .....	514
12.3.7	app_weld_set_interface_eip_r2m_option(opt1=[0,0,0,0,0,0,0,0], opt2=[0,0,0,0,0,0,0,0], opt3=[0,0,0,0,0,0,0,0], opt4=[0,0,0,0,0,0,0,0], opt5=[0,0,0,0,0,0,0,0], opt6=[0,0,0,0,0,0,0,0],...) .....	518

---

12.3.8	app_weld_set_interface_eip_m2r_process(current_flow=[0,0,0,0,0,0,0,0], process_active=[0,0,0,0,0,0,0,0], main_current=[0,0,0,0,0,0,0,0], machine_ready=[0,0,0,0,0,0,0,0], comm_ready=[0,0,0,0,0,0,0,0]) .....	522
12.3.9	app_weld_set_interface_eip_m2r_monitoring(welding_voltage=[0,0,0,0,0,0,0,0], welding_current=[0,0,0,0,0,0,0,0], wire_feed_speed=[0,0,0,0,0,0,0,0], wire_stick=[0,0,0,0,0,0,0,0], error=[0,0,0,0,0,0,0,0] ...) .....	525
12.3.10	app_weld_set_interface_eip_m2r_other(opt1=[0,0,0,0,0,0,0,0], opt2=[0,0,0,0,0,0,0,0], opt3=[0,0,0,0,0,0,0,0], opt4=[0,0,0,0,0,0,0,0], opt5=[0,0,0,0,0,0,0,0], opt6=[0,0,0,0,0,0,0,0], opt7=[0,0,0,0,0,0,0,0],...) .....	529
12.3.11	app_weld_set_weld_cond_digital(flag_dry_run=0, vel_target=0, vel_min=0, vel_max=0, welding_mode=0, s_2t=0, pulse_mode=0, wm_opt1=0, simulation=0, ts_opt1=0, ts_opt2=0,...) .....	533
12.3.12	app_weld_adj_welding_cond_digital(flag_reset=None, f_target=None, vel_target=None, wv_offset=None, wv_width_ratio=None, dynamic_cor=None, voltage_cor=None, job_number=None, synergic_id=None) .....	537
12.3.13	app_weld_get_welding_cond_digital() .....	541
12.3.14	app_weld_enable_analog(ch_v_out=[1,0], spec_v_out=[0,0,0], ch_f_out=[2,0], spec_f_out=[0,0,0], ch_v_in=[1,0], spec_v_in=[0,0,0], ch_c_in=[2,0], spec_c_in=[0,0,0], ch_arc_on=1, ch_gas_on=2, ch_inching_fwd=3, ch_inching_bwd=4, ...)5 45	545
12.3.15	app_weld_disable_analog() .....	550
12.3.16	app_weld_set_weld_cond_analog(flag_dry_run=0, v_target=0, f_target=0, vel_target=0, vel_min=0, vel_max=0, weld_proc_param=[0.2,0.2,0.5,0.5,0.5,0.2,0.2,0.5,0.5]) .....	551
12.3.17	app_weld_adj_welding_cond_analog(flag_reset=0, v_target=None, f_target=None, vel_target=None, wv_offset=None, wv_width_ratio=None) .....	555
12.3.18	app_weld_get_welding_cond_analog() .....	557
12.3.19	app_weld_weave_cond_trapezoidal(wv_offset=[0,0], wv_ang=0, wv_param=[0,1.5,0,-1.5,0.3,0.1,0.3,0.3,0.1,0.3]) .....	560
12.3.20	app_weld_weave_cond_zigzag(wv_offset=[0,0], wv_ang=0, wv_param=[3,0.6]) .....	562
12.3.21	app_weld_weave_cond_circular(wv_offset=[0,0], wv_ang=0, wv_param=[3,3,0.3,0.3]) .....	564
12.3.22	app_weld_weave_cond_sinusoidal(wv_offset=[0,0], wv_ang=0, wv_param=[3,0.6])....	566

## 13 A-Series 전용 명령어 ..... 569

13.1	컨트롤러 .....	569
13.1.1	get_function_input(index).....	569

# 1 머리말

본 매뉴얼은 총 13장으로 구성되어 있으며 2장부터 12장까지는 M, H, A 시리즈 모델에 공통으로 적용되는 명령어이며, 13장은 A 시리즈 모델에만 적용되는 명령어입니다.

본 매뉴얼의 내용은 작성된 시점을 기준으로 하며, 제품에 대한 정보는 사용자에게 사전 고지 없이 변경될 수 있습니다.

개정된 매뉴얼에 대한 상세 정보는 Robot LAB (<https://robotlab.doosanrobotics.com/>)에서 확인하십시오.

## 1.1 저작권

본 매뉴얼의 모든 내용과 도안에 대한 저작권 및 지적재산권은 두산로보틱스에 있습니다. 따라서 두산로보틱스의 서면 허가 없이 사용, 복사, 유포하는 어떠한 행위도 금지됩니다. 또한 특허권을 오용하거나 변용하는데 따르는 책임은 전적으로 사용자에게 있습니다.

본 매뉴얼은 신뢰할 수 있는 정보지만 오류 또는 오탈자로 인한 손실에 대해 어떠한 책임도 지지 않습니다. 제품 개선에 따라 매뉴얼에 포함된 정보는 예고 없이 변경될 수 있습니다.

© Doosan Robotics Inc., All rights reserved

## 2 DRL Basic Syntax

### 2.1 기본 문법

**⚠ 주의**

DRL의 문법은 python 과 동일하지만, Python의 모든 문법과 기능을 포함하지 않습니다.  
본 매뉴얼에 기술된 내용만 지원합니다.

#### 2.1.1 들여쓰기

##### 기능

각 코드 블록과의 구분을 위해 사용합니다. 들여쓰기를 준수하지 않을 시 오류가 발생합니다.

- 들여 쓰기는 각 코드 블록을 구분하는 데 사용됩니다.
- 들여 쓰기에는 2 칸, 4 칸 또는 탭 문자를 사용할 수 있습니다.
- 블록의 경우 동일한 유형의 들여 쓰기가 사용되어야합니다

##### 예제

```

1  코드블럭1
2  [TAB]코드블럭2
3
4  예)
5  if x == 3:
6    y += 1
7
8  # No Error
9  def fnSum(x,y):
10   [공백4]sum = x + y
11   [공백4]return sum
12
13 # No Error
14 def fnSubtract(x,y):
15   [TAB]diff = x - y
16   [TAB]return diff
17
18 # 들여쓰기 오류
19 def fnProduct(x,y):
20   [공백4]product = x * y
21   [TAB]return product

```

## 2.1.2 주석

### 기능

해당 코드를 부가적으로 설명하기 위해 사용합니다. 주석은 실제 코드 처리 과정에서 제외되기 때문에 소스 코드에 영향을 주지 않습니다.

# 이후로는 주석으로 인식합니다. ””로 시작해서 ””로 끝나는 블록은 주석으로 인식합니다.

- 주석은 코드에 대한 추가 설명을 제공하는 데 사용됩니다. 주석은 코드 처리에서 제외되므로 소스 코드에 영향을 미치지 않습니다.
- """다음의 명령문은 주석으로 인식됩니다.
- ””로 시작하고 ””로 끝나는 블록은 주석으로 인식됩니다.

### 예제

```

1 # 주석 예1
2 """
3     주석 예2"""

```

## 2.2 변수

### 2.2.1 변수명

#### 기능

- 자료의 값을 나타내기 위해 사용하며 문자, 숫자, 밑줄(\_)로 구성할 수 있습니다. 문자, 숫자, 밑줄(\_)로 구성합니다(첫 글자는 숫자 불가).
- 대소문자를 구분합니다.
- 변수명을 예약어나 DRL 명령어 이름과 동일하게 사용하면 인터프리터에 치명적인 에러가 발생합니다. 이를 피하기 위해서 가급적 “var\_” 이란 prefix를 사용하여 변수명을 사용하시기 바랍니다.

#### 주의

하기 예약어들을 변수명이나 함수명으로 절대로 사용하지 마십시오.

<b>and</b>	<b>assert</b>	<b>break</b>	<b>class</b>	<b>continue</b>
<b>def</b>	<b>del</b>	<b>elif</b>	<b>else</b>	<b>except</b>
<b>exec</b>	<b>finally</b>	<b>for</b>	<b>from</b>	<b>global</b>

<b>if</b>	<b>import</b>	<b>in</b>	<b>is</b>	<b>lambda</b>
<b>list</b>	<b>not</b>	<b>open</b>	<b>or</b>	<b>pass</b>
<b>print</b>	<b>raise</b>	<b>return</b>	<b>try</b>	<b>while</b>
<b>with</b>	<b>yield</b>	<b>select</b>		

### 예제

```

1 friend = 10
2 Friend = 1
3 _myFriend = None
4
5 Pass = 10 # Syntax error
6 movej = 0 # movej는 DRL 명령어 이름으로 변수로 사용하면 안됩니다

```

## 2.2.2 수치

### 기능

int, float, complex로 수치를 입력합니다.

### 예제

```

1 10, 0x10, 0o10, 0b10
2 3.14, 314e-2
3 x = 3-4j
4
5 int_value = 10
6 hexa_value = 0x10
7 octa_value = 0o10
8 binary_value = 0b10
9 double_value = 3.14
10 double_value = 314e-2
11 complex_value = 3-4j

```

## 2.2.3 문자

### String

### 기능

모든 문자열은 기본적으로 유니코드입니다.

- Escape characters
  - \n: New line
  - \t: Tab
  - \r: Carriage return
  - \0: Null string
  - \\: back slash(\) in string
  - \': single quote mark in string
  - \": double quote mark in string
- String concatenation: "py"+ "tyon" → "python"
- String repetition: "py" \* 3 → "pypypy"
- String indexing: "python" [0] → "p"
- String slicing: "python" [1:4] → "yth"

## 예제

```

1  "string1"
2  'string2'
3
4  tp_log("st"+"ring")
5  #expected result: string
6  tp_log("str" * 3)
7  #expected result: strstrstr
8  tp_log("line1\nline2")
9  #expected result: line1
10 # line2
11 tp_log("\\"string\\\"")
12 #expected result: "string"
13 tp_log("str"[0])
14 #expected result: s
15 tp_log("string"[1:3])
16 #expected result: tr

```

+, \*

## 예제

1	"Doosan"+ "Robotics" → "DoosanRobotics"
2	"Doo" * 3 → "DooDooDoo"

## 인덱싱 & 슬라이싱

### 예제

```

1 " Doosan" [0] → "D"
2 " Doosan" [1:4] → "oos"

```

## 2.2.4 list

### 기능

- 값들의 나열로, 각 item은 변경 가능하며 순서가 있습니다.
- 인덱싱 및 슬라이싱이 가능합니다.
- append, insert, extend, +연산자
- count, remove, sort 연산자

### 예제

```

1 colors = ["red", "green", "blue"]
2 tp_log(colors[0]+","+colors[1]+","+colors[2])
    #expected print result: red,green,blue
4
5 numbers = [1, 3, 5, 7, 9]
6 sum = 0
7 for number in numbers:
8     sum += number
9 tp_log( str(sum) )
    #expected result: 25
10

```

## 2.2.5 tuple

### 기능

list와 비슷하나, 읽기 전용으로 속도가 빠릅니다.

### 예제

```

1 colors = ("red", "green", "blue")
2 numbers = (1, 3, 5, 7, 9)
3
4 def fnMinMax(numbers):
5     numbers.sort()

```

```

6   return (numbers[0], numbers[-1])
7   minmax = fnMinMax([4,1,2,9,5,7])
8   tp_log("Min Value= " + str(minmax[0]))
9   #expected result: Min Value = 1
10  tp_log("Max Value= " + str(minmax[1]))
11  #expected result: Max Value = 9

```

## 2.2.6 dictionary

### 기능

Key와 Value를 지정하여 값을 나열합니다.

### 예제

```

1 d = dict(a = 1, b = 3, c = 5)
2
3 colors = dict()
4 colors["cherry"] = "red"
5
6 ages = {'Kim':35, 'Lee':38, 'Chang':37}
7 tp_log("Ages of Kim = " + str(ages['Kim']))
8 #expected print result: Ages of Kim = 35

```

## 2.3 함수

### 2.3.1 함수 문법

#### 기능

- 선언: def로 시작하고 콜론(:)으로 끝냅니다.
- 함수의 시작과 끝은 코드의 들여쓰기로 구분합니다.
- interface/implementation을 따로 구분하지 않습니다. 단, 사용 전에 미리 정의되어야 합니다.
- 함수명을 예약어나 DRL 명령어 이름과 동일하게 사용하면 인터프리터에 치명적인 에러가 발생합니다.  
이를 피하기 위해서 가급적 “fn\_” 이란 prefix를 사용하여 함수 이름을 사용하시기 바랍니다

#### 주의

하기 예약어들을 변수명이나 함수명으로 절대로 사용하지 마십시오.

<b>and</b>	<b>assert</b>	<b>break</b>	<b>class</b>	<b>continue</b>
<b>def</b>	<b>del</b>	<b>elif</b>	<b>else</b>	<b>except</b>

<b>exec</b>	<b>finally</b>	<b>for</b>	<b>from</b>	<b>global</b>
<b>if</b>	<b>import</b>	<b>in</b>	<b>is</b>	<b>lambda</b>
<b>list</b>	<b>not</b>	<b>open</b>	<b>or</b>	<b>pass</b>
<b>print</b>	<b>raise</b>	<b>return</b>	<b>try</b>	<b>while</b>
<b>with</b>	<b>yield</b>	<b>select</b>		

## 문법

```
def <함수명>(인수1, 인수2, ... 인수N):
    <구문> ...
    return <반환값>
```

```

1   #예)
2   def fn_Times(a, b):
3       return a * b
4
5   fn_Times (10, 10)
6
7   def fn_Times(a, b):
8       return a * b
9
10  tp_log(str(fn_Times(10, 5)))
11  #expected result: 50
12
13  def movej(): #movej는 DRL 명령어로 함수명으로 사용하면 안 됩니다.
14      return 0

```

## 2.3.2 스코핑 룰

### 기능

- 함수내에 지역변수에 해당하는 이름의 값이 없을 경우 LGB 규칙에 기반하여 이름을 찾을 수 있습니다.
  - 이름공간 : 변수의 이름이 저장되어 있는 장소
  - Local scope : 함수 내부의 이름 공간, 지역 영역
  - Global scope : 함수 밖의 영역, 전역 영역
  - Built-in scope : 파이썬 자체에서 정의한 내용에 대한 영역, 내장 영역
  - LGB 규칙: 변수 이름을 찾는 순서 local → global → built-in

### 예제

```
1   # Error: can't find simple_pi in circle_area
```

```

2 simple_pi = 3.14
3 def circle_area(r):
4     return r*r*simple_pi
5
6 # simple_pi should be declared as global if it is used in circle_area_ok
7 def circle_area_ok(r):
8     global simple_pi
9     return r*r*simple_pi
10
11 tp_log(str(circle_area(3.0)))
12 #expected result: 28.26

```

### 2.3.3 인수 모드

#### 기능

기본 인수 값, 키워드 인수, 가변 인수를 사용합니다.

#### 예제

```

1 def fn_Times(a = 10, b = 20):
2     return a * b
3
4 #Example - Default parameter value
5 tp_log(str(fn_Times(5)))
6 #expected result: 100
7
8 #Example - Keyword parameter
9 tp_log(str(fn_Times(b=5)))
10 #expected result: 50
11 tp_log(str(fn_Times(a=5, b=5)))
12 #expected result: 25
13
14 #Example - arbitrary parameter
15 def fn_myUnion(*args):
16     for arg in args:
17         tp_log(str(arg))
18         fn_myUnion("red", 1)
19 #expected print result: red
20 # 1

```

#### 예제 - 기본 인수 값

```

1 def fn_Times(a = 10, b = 20):
2     return a * b
3

```

	4 fn_Times(5)
--	---------------

### 예제 - 키워드 인수

	1 def fn_Times(a = 10, b = 20) 2       return a * b 3 4 fn_Times(a=5, b=5)
--	---

### 예제 - 가변 인수

	1 def fn_myUnion(*ar) 2       ..... 3 4 fn_myUnion("red", "white", "black")
--	--

## 2.4 제어문

### 2.4.1 pass

#### 기능

어떠한 동작을 실행하지 않을 때 사용합니다.

#### 예제

	1 while True: 2     pass #pass means empty statement, so while statement continues to run. 3     tp_log("This line never reached")
--	--

### 2.4.2 if

#### 기능

조건문 함수로 if 문의 조건식에 참인가 거짓인가에 따라 elif와 else를 사용할 수 있습니다.

#### 문법

**if <조건식>:**

<구문>

```
if <조건식1>:
```

<구문1>

```
elif <조건식2>:
```

<구문2>

```
else:
```

<구문3>

### 예제 - if, elif, else

1	numbers = [2,5,7]
2	for number in numbers:
3	if number%2==0:
4	tp_log(str(number) + " is even")
5	else:
6	tp_log (str(number) + " is odd")
7	#expected result:
8	#2 is even
9	#5 is odd
10	#7 is odd

## 2.4.3 while

### 기능

조건문 함수로 참과 거짓에 따라 반복작업을 실행합니다.

### 문법

```
while <조건식>:
```

<구문>

### 예제

1	sum = 0
2	cnt = 1
3	while cnt < 10:
4	sum = sum+cnt
5	cnt = cnt+1
6	tp_log("sum = " + str(sum))
7	#expected result:

8	#sum = 45
---	-----------

## 2.4.4 for

### 기능

range로 반복 범위를 설정하여 반복작업을 실행합니다.

### 문법

**for <item> in <sequence>형 객체 S:**

<구문>

### 예제

```

1 x=0
2 for i in range(0, 3): # i는 0 -> 1 -> 2
3 x= x + 1
4
5 sum = 0
6 for i in range(0, 10):
7     sum = sum + i
8     tp_log("sum = " + str(sum))
9 #expected result:
10 #sum = 45

```

## 2.4.5 break

### 기능

반복문 내부 블록을 빠져나올 때 사용합니다.

### 예제

```

1 x =0
2 while True:
3     x = x + 1
4     if x > 10:
5         break
6
7 sum = 0;cnt = 0
8 while True:
9     if cnt > 9:
10        break

```

```

11 sum = sum + cnt
12 cnt = cnt+1
13 tp_log("sum = " + str(sum))
14 #expected print result:
15 #sum = 45

```

## 2.4.6 continue

### 기능

반복문 내부 블록을 더 이상 수행하지 않고 반복문 시작 지점으로 이동할 때 사용합니다.

### 예제

```

1 #<ex> 1
2 x=0
3 y=0
4 while True:
5     x = x + 1
6     if x > 10:
7         continue
8     y += 100
9
10    #<ex> 2
11    sum = 0
12    for i in range(0, 10):
13        if i%2==0:
14            continue
15        sum = sum + i
16    tp_log("sum of odd numbers = " + str(sum))
17    #sum of odd numbers = 25

```

## 2.4.7 반복문의 else

### 기능

반복문 수행 도중 break 함수로 인하여 반복문이 종료되지 않고 끝까지 수행되었을 때 else 블록이 실행됩니다

### 예제

```

1 L = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 }
2
3 for i in L:
4     if i % 2 == 0:
5         continue
6     else:

```

7 tp\_log("exit without break")

## 3 모션 관련 명령어

### 3.1 위치 생성

#### 3.1.1 posj(J1=0, J2=0, J3=0, J4=0, J5=0, J6=0)

##### 기능

조인트 공간 각도를 좌표값으로 지정합니다.

##### 인수

인수명	자료형	기본값	설명
J1	float list posj	0	1축 angle 또는 angle list 또는 posj
J2	float	0	2축 angle
J3	float	0	3축 angle
J4	float	0	4축 angle
J5	float	0	5축 angle
J6	float	0	6축 angle

##### 리턴

Posj

##### 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시

## 예제

```

1 q1 = posj()                                # q1=posj(0,0,0,0,0,0)
2
3 q2 = posj(0, 0, 90, 0, 90, 0)
4
5 q3 = posj([0, 30, 60, 0, 90, 0])          # q3=posj(0,30,60,0,90,0)

```

## 관련 명령어

- movej()(p. 57)
- amovej()(p. 100)
- movesj()(p. 77)
- amovesj()(p. 113)

### 3.1.2 posx(x=0, y=0, z=0, A=0, B=0, C=0, ori\_type=None, sol=None, turn=None)

#### 기능

작업 공간을 좌표값으로 지정합니다.

#### **i 알아두기**

- SW 버전 V3.2 부터, 다양한 Orientation type 입력을 지원합니다.

#### 인수

인수명	자료형	기본값	설명
x	float list posx	0	x position 또는 position list 또는 posx
y	float	0	y position
z	float	0	z position
A	float	0	A orientation
B	float	0	B orientation

인수명	자료형	기본값	설명
C	float	0	C orientation
ori_type	int	None	orientation type <ul style="list-style-type: none"> <li>• DR_ELR_ZYZ(if None): Euler Angles(z-y'-z'', in degrees)</li> <li>• DR_ELR_ZYX: Euler Angles(z-y'-x'', in degrees)</li> <li>• DR_ELR_XYZ: Euler Angles(x-y'-z'', in degrees)</li> <li>• DR_FIX_XYZ: Fixed Angles(x-y-z, in degrees)</li> <li>• DR_ROTVEC: 3D rotation vector (angle/axis representation)</li> <li>• DR_QUAT: unit quaternion(x, y, z, w)</li> </ul>
sol	int	None	solution space <ul style="list-style-type: none"> <li>• None: robot configuration 자동 선택 - 추후 지원 예정</li> <li>• 0~7: robot configuration 지정 (ikin_norm 함수 참조)</li> </ul>
turn	int	None	joint multi turn - 추후 지원 예정 <ul style="list-style-type: none"> <li>• None: joint multi turn 자동 선택</li> </ul>

## 리턴

값	설명
posx	task space point

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시

## 예제

```

1 movej([0,0,90,0,90,0], v=10, a=20)
2
3 x2 = posx(400, 300, 500, 0, 180, 0)          # Euler ZYZ
4
5 x3 = posx([350, 350, 450, 0, 180, 0])        # x3=posx(350, 350, 450, 0,
6 180, 0), Euler ZYZ

```

```

7   x4 = posx(x2)                               # x4=posx(400, 300, 500, 0,
8   180, 0), Euler ZYZ
9
10  x5 = posx(400, 300, 500, 0, 180, 0, DR_ELR_ZYX) # Euler ZYX
11
12  x6 = posx([400, 300, 500, 0, 0, 0, 1], ori_type=DR_QUAT) # Quaternion,
13   only list input available
14
15  x6.ori_type = DR_ROTVEC                      # Transform to rot vec
16   (setter method)
17
18  x7 = posx(x6, ori_type=DR_FIX_XYZ)          # Constructor as the other
19   orientation type
20
21  moveL(x2, v=100, a=200)

```

## 관련 명령어

- [moveL\(\)](#)(p. 62)
- [moveC\(\)](#)(p. 71)
- [moveJx\(\)](#)(p. 67)
- [amoveL\(\)](#)(p. 103)
- [amoveC\(\)](#)(p. 109)
- [amoveJx\(\)](#)(p. 106)

### 3.1.3 trans(pos, delta, ref, ref\_out, ori\_type\_out)

#### 기능

- ref 좌표계 기준으로 정의된 pos(포즈)를 동일 좌표계를 기준으로 delta만큼 이동/회전하여 ref\_out 좌표계 기준으로 변환한 후 리턴합니다.
- 단, ref 인자가 Tool Coordinate인 경우, ref\_out인자는 무시되며 pos와 동일한 좌표계 기준의 결과를 리턴합니다.

#### 인수

인수명	자료형	기본값	설명
pos	posx	-	posx 또는 position list
	list (float[6])		
delta	posx	-	posx 또는 position list
	list (float[6])		

인수명	자료형	기본값	설명
ref	int	None	<p>reference coordinate</p> <ul style="list-style-type: none"> <li>• DR_BASE : base coordinate</li> <li>• DR_WORLD : world coordinate</li> <li>• DR_TOOL : tool coordinate</li> <li>• user coordinate: 사용자 정의</li> </ul>
ref_out	int	DR_BASE	<p>reference coordinate</p> <ul style="list-style-type: none"> <li>• DR_BASE : base coordinate</li> <li>• DR_WORLD : world coordinate</li> <li>• user coordinate: 사용자 정의</li> </ul>
ori_type_out	int	None	<p>output orientation type</p> <ul style="list-style-type: none"> <li>• None: pos 인자의 orientation type을 따름.</li> <li>• DR_ELR_ZYZ: Euler Angles(z-y'-z'', in degrees)</li> <li>• DR_ELR_ZYX: Euler Angles(z-y'-x'', in degrees)</li> <li>• DR_ELR_XYZ: Euler Angles(x-y'-z'', in degrees)</li> <li>• DR_FIX_XYZ: Fixed Angles(x-y-z, in degrees)</li> <li>• DR_ROTVEC: 3D rotation vector (angle/axis representation)</li> <li>• DR_QUAT: unit quaternion(x, y, z, w)</li> </ul>

## 리턴

값	설명
posx	task space point

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시

예외	설명
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 p0 = posj(0,0,90,0,90,0)
2
3 movej(p0, v=30, a=30)
4
5
6
7 x1 = posx(200, 200, 200, 0, 180, 0)
8
9 delta = [100, 100, 100, 0, 0, 0]
10
11 x2 = trans(x1, delta, DR_BASE, DR_BASE)           # translation of x1 with
delta in base coordinates
12
13 x3 = trans(x1, delta, ori_type_out=DR_ELR_ZYX) # transform to euler zyx
14
15
16 x1_base = posx(500, 45, 700, 0, 180, 0)
17
18 x4 = trans(x1_base, [10, 0, 0, 0, 0, 0], DR_TOOL)
19
20 movel(x4, v=100, a=100, ref=DR_BASE)
21
22
23
24 uu1 = [1, 1, 0]
25
26 vv1 = [-1, 1, 0]
27
28 pos = posx(559, 34.5, 651.5, 0, 180.0, 0)
29
30 DR_userTC1 = set_user_cart_coord(uu1, vv1, pos) #사용자 정의 좌표계 등록
31
32 x1_userTC1 = posx(30, 20, 100, 0, 180, 0)        #사용자좌표계 상의 posx
33
34 x9 = trans(x1_userTC1, [0, 0, 50, 0, 0, 0], DR_userTC1, DR_BASE)
35
36 movel(x9, v=100, a=100, ref=DR_BASE)

```

## 관련 명령어

- posx(x=0, y=0, z=0, A=0, B=0, C=0, ori\_type=None, sol=None, turn=None)(p. 30)
- addto(pos, add\_val=None)(p. 41)

### 3.1.4 posb(seg\_type, posx1, posx2=None, radius=0)

#### 기능

- 정속 블렌딩 모션(moveb, amoveb)의 입력 인자로, 각 경유점의 좌표와 단위 경로 형태(라인 또는 원호)의 정보를 갖는 posb는 블랜딩되는 trajectory의 단위 세그먼트 객체를 정의합니다.
- seg\_type이 line인 경우(DR\_LINE)는 posx1만 입력, circle인 경우(DR\_CIRCLE)는 posx2까지 입력합니다. radius는 이어지는 segment와의 blending 반경을 설정합니다.

#### 인수

인수명	자료형	기본값	설명
seg_type	Int	-	DR_LINE DR_CIRCLE
posx1	posx	-	1 <sup>st</sup> task posx
posx2	posx	-	2 <sup>nd</sup> task posx
radius	float	0	Blending radius [mm]

#### 리턴

Posb

#### 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시

#### 예제

```

1 q0 = posj(0, 0, 90, 0, 90, 0)
2 movej(q0, vel=30, acc=60)
3
4 x0 = posx(564, 34, 690, 0, 180, 0)
5
6 movel(x0, vel=200, acc=400)      # 시작위치로 이동
7

```

```

8
9
10
11 x1 = posx(564, 200, 690, 0, 180, 0)
12
13 seg1 = posb(DR_LINE, x1, radius=40)
14
15 x2 = posx(564, 100, 590, 0, 180, 0)
16
17 x2c = posx(564, 200, 490, 0, 180, 0)
18
19 seg2 = posb(DR_CIRCLE, x2, x2c, radius=40)
20
21 x3 = posx(564, 300, 490, 0, 180, 0)
22
23 seg3 = posb(DR_LINE, x3, radius=40)
24
25 x4 = posx(564, 400, 590, 0, 180, 0)
26
27 x4c = posx(564, 300, 690, 0, 180, 0)
28
29 seg4 = posb(DR_CIRCLE, x4, x4c, radius=40)
30
31 x5 = posx(664, 300, 690, 0, 180, 0)
32
33 seg5 = posb(DR_LINE, x5, radius=40)
34
35 x6 = posx(564, 400, 690, 0, 180, 0)
36
37 x6c = posx(664, 500, 690, 0, 180, 0)
38
39 seg6 = posb(DR_CIRCLE, x6, x6c, radius=40)
40
41 x7 = posx(664, 400, 690, 0, 180, 0)
42
43 seg7 = posb(DR_LINE, x7, radius=40)
44
45 x8 = posx(664, 400, 590, 0, 180, 0)
46
47 x8c = posx(564, 400, 490, 0, 180, 0)
48
49 seg8 = posb(DR_CIRCLE, x8, x8c, radius=0)          # 마지막 radius는 0이어야 함
50
51     # 만약 0이 아닌 경우 0으로 처리됨
52
53
54
55 b_list = [seg1, seg2, seg3, seg4, seg5, seg6, seg7, seg8]
56
57
58 moveb(b_list, vel=200, acc=400)
59

```

## 관련 명령어

- [posx\(x=0, y=0, z=0, A=0, B=0, C=0, ori\\_type=None, sol=None, turn=None\)\(p. 30\)](#)
- [moveb\(\)\(p. 85\)](#)
- [amoveb\(\)\(p. 119\)](#)

### 3.1.5 fkin(pos, ref, ori\_type)

#### 기능

조인트 공간에서 조인트각도 또는 이에 상응하는 자료형(float[6])을 입력받아 ref 좌표계 기준의 TCP의 포즈(태스크 공간의 위치 및 자세)를 리턴합니다.

#### 인수

인수명	자료형	기본값	설명
pos	posj	-	posj 또는 position list
	list (float[6])		
ref	int	DR_BASE	reference coordinate <ul style="list-style-type: none"> <li>DR_BASE : base coordinate</li> <li>DR_WORLD : world coordinate</li> </ul>
ori_type	int	DR_ELR_ZYZ	orientation type <ul style="list-style-type: none"> <li>DR_ELR_ZYZ: Euler Angles(z-y'-z'', in degrees)</li> <li>DR_ELR_ZYX: Euler Angles(z-y'-x'', in degrees)</li> <li>DR_ELR_XYZ: Euler Angles(x-y'-z'', in degrees)</li> <li>DR_FIX_XYZ: Fixed Angles(x-y-z, in degrees)</li> <li>DR_ROTVEC: 3D rotation vector (angle/axis representation)</li> <li>DR_QUAT: unit quaternion(x, y, z, w)</li> </ul>

#### 리턴

값	설명
posx	Task space point

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시

## 예제

```

1 q1 = posj(0, 0, 90, 0, 90, 0)
2
3 movej(q1,v=10,a=20)
4
5 q2 = posj(30, 0, 90, 0, 90, 0)
6
7 x2 = fkin(q2, DR_WORLD) # x2: 관절값 q2에 대응하는 로봇끝단(TCP)의 공간좌표
8
9 movel(x2,v=100,a=200,ref=DR_WORLD) # x2로 직선모션 수행
10
11
12 x1 = fkin(q1, DR_WORLD, DR_ELR_XYZ) # x1: 관절값 q1에 대응하는 로봇끝단(TC)의 공간
13 좌표 (orientation type: euler xyz)
14 movel(x1,v=100,a=200,ref=DR_WORLD) # x1로 직선모션 수행

```

## 관련 명령어

- set\_tcp(name)(p. 52) : 터치펜던트에 등록한 이름의 tcp 정보가 fkin연산 시 반영
- posj(J1=0, J2=0, J3=0, J4=0, J5=0, J6=0)(p. 29)
- posx(x=0, y=0, z=0, A=0, B=0, C=0, ori\_type=None, sol=None, turn=None)(p. 30)

### 3.1.6 ikin(pos, sol\_space, ref, ref\_pos\_opt, iter\_threshold)

#### 기능

작업 공간내 기준 좌표계(ref)의 로봇 포즈에 상응하는 8개의 관절형상 중 지정한 관절형상(sol\_space)에 해당하는 관절각도를 리턴합니다. 관절각도는 옵션(ref\_pos\_opt)에 따라 기준 각도에 가장 가까운 값을 리턴합니다. 리턴 값 status를 통해 현재 로봇 포즈가 동작 영역을 벗어나는지 손목 축 특이점 영역에 있는지 확인 할 수 있습니다.

#### 알아두기

- SW 버전 V2.9 이후, 본 기능을 사용할 때 로봇의 정확도가 보정이 된 경우 정확도 보정 알고리즘이 동작합니다. 정확도 수준은 옵션(iter\_threshold)에 따라 조정이 가능합니다. 또한 로봇 자세에 따라서 최대 0.1초의 지연이 있을 수 있습니다.
- 로봇 각 모델 별로 알려진 보정 전 DH 파라미터 값에 해당하는 역기구학 해를 구하기 위해서는 ikin\_norm() 명령어를 참조 하시기 바랍니다.

#### 주의

- res\_pos\_opt 파라미터를 입력하지 않을 경우 pos 만 반환됩니다. ref\_pos\_opt 파라미터를 입력하지 않은 상태에서 ref, status = ikin(...) 형태로 사용할 경우 Runtime error가 발생할 수 있습니다.

#### 인수

인수명	자료형	기본값	설명
pos	posx	-	posx 또는 position list
	list (float[6])		
sol_space	int	-	solution space <ul style="list-style-type: none"> <li>None: 3.2부터는 사용하지 않음.</li> <li>0~7: ~2.x drl 호환용.</li> </ul>
ref	int	DR_BASE	reference coordinate <ul style="list-style-type: none"> <li>DR_BASE : base coordinate</li> <li>DR_WORLD : world coordinate</li> </ul>
ref_pos_opt	int	0	멀티턴 해 생성시 옵션에 따른 기준 위치에 가까운 해 선택 <ul style="list-style-type: none"> <li>0 : posj(0,0,0,0,0,0) 조인트 각도 기준</li> <li>1 : 현재 조인트 각도 기준</li> </ul>

인수명	자료형	기본값	설명
iter_threshold	list (float[2])	0.005	정확도가 보정 된 경우, 정확도 보정 알고리즘의 수준 TCP position (X, Y, Z)의 norm 값 [mm]
		0.01	정확도가 보정 된 경우, 정확도 보정 알고리즘의 수준 TCP orientation (A, B, C)의 norm 값 [deg]

### Solution Space w.r.t. Robot Configuration

Solution space	Binary	Shoulder	Elbow	Wrist
0	000	Lefty	Below	No Flip
1	001	Lefty	Below	Flip
2	010	Lefty	Above	No Flip
3	011	Lefty	Above	Flip
4	100	Righty	Below	No Flip
5	101	Righty	Below	Flip
6	110	Righty	Above	No Flip
7	111	Righty	Above	Flip

### 리턴

값	설명
posj	Joint space point
status	<ul style="list-style-type: none"> <li>• 0 : 조인트 값 정상 반환</li> <li>• 1 : 동작 영역을 벗어남</li> <li>• 2 : 손목 축 특이점</li> </ul>

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 ## 2.x 호환 코드
2 x1 = posx(370.9, 719.7, 651.5, 90, -180, 0)
3 q1, status = ikin(x1, 2, DR_BASE, ref_pos_opt=0) # 로봇끝단의 좌표가 x1이 되는 관
4 절각 q1 (8가지 경우 중 첫번째), posj(0,0,0,0,0,0) 조인트 각도 기준
5 # q1=posj(60.3, 81.0, -60.4, -0.0, 159.4, -29.7) (M1013, tcp=(0,0,0)경우)
6 movej(q1,v=10,a=20)
7
8 ## 3.2~ 코드
9 x1 = posx(370.9, 719.7, 651.5, 90, -180, 0, sol=2)
10 q1, status = ikin(x1, ref=DR_BASE, ref_pos_opt=0) # 로봇끝단의 좌표가 x1이 되는 관
11 절각 q1 (8가지 경우 중 첫번째), posj(0,0,0,0,0,0) 조인트 각도 기준
    # q1=posj(60.3, 81.0, -60.4, -0.0, 159.4, -29.7) (M1013, tcp=(0,0,0)경우)
    movej(q1,v=10,a=20)

```

## 관련 명령어

- set\_tcp(name)(p. 52)
- posj(J1=0, J2=0, J3=0, J4=0, J5=0, J6=0)(p. 29)
- posx(x=0, y=0, z=0, A=0, B=0, C=0, ori\_type=None, sol=None, turn=None)(p. 30)

### 3.1.7 addto(pos, add\_val=None)

#### 기능

posj의 각 관절 값에 add\_val 만큼 추가하여 새로운 posj 객체를 생성합니다.

## 인수

인수명	자료형	기본값	설명
pos	posj	-	posj 또는 position list
	list (float[6])		
add_val	list (float[6])	None	Position에 추가할 add 값 목록 • None 또는 [] 일 시에는, 값을 추가 안 함

## 리턴

값	설명
posj	Joint space point

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시

## 예제

```

1 q1 = posj(10, 20, 30, 40, 50, 60)
2 movej (q1, v=10, a=20)
3
4 q2 = addto(q1, [0, 0, 0, 0, 45, 0])
5
6 movej (q2, v=10, a=20) # 로봇이 관절(10, 20, 30, 40, 95, 60)로 이동합니다.
7
8 q3 = addto(q2, [])
9
10 q4 = addto(q3)
11

```

## 관련 명령어

- posj(J1=0, J2=0, J3=0, J4=0, J5=0, J6=0)(p. 29)

### 3.1.8 ikin\_norm(pos, sol\_space, ref, ref\_pos\_opt)

#### 기능

작업 공간내 기준 좌표계(ref)의 로봇 포즈에 상응하는 8개의 관절형상 중 지정한 관절형상(sol\_space)에 해당하는 관절각도를 리턴합니다. 관절각도는 옵션(ref\_pos\_opt)에 따라 기준 각도에 가장 가까운 값을 리턴합니다. 리턴 값 status를 통해 현재 로봇 포즈가 동작 영역을 벗어나는지 손목 축 특이점 영역에 있는지 확인 할 수 있습니다.

#### 알아두기

- SW 버전 V2.9 이후 로봇은 정확도 향상을 위해 DH 파라미터를 보정할 수 있으며, V2.10.1 버전 이후 출시되는 로봇은 정확도 향상을 위해 DH 파라미터가 보정되어 출시 됩니다.
- 해당 명령어를 통해 로봇 각 모델 별로 알려진 보정 전 DH 파라미터 값에 해당하는 역기구학 해를 구할 수 있습니다.
- 보정된 DH 파라미터에 해당 하는 역기구학 해를 구하기 위해서는 ikin() 명령어를 참조 하시기 바랍니다.

#### 인수

인수명	자료형	기본값	설명
pos	posx	-	posx 또는 position list
	list (float[6])		
sol_space	int	-	solution space <ul style="list-style-type: none"> <li>• None: 3.2부터는 사용하지 않음.</li> <li>• 0~7: ~2.x drl 호환용.</li> </ul>
ref	int	DR_BASE	reference coordinate <ul style="list-style-type: none"> <li>• DR_BASE : base coordinate</li> <li>• DR_WORLD : world coordinate</li> </ul>
ref_pos_opt	int	0	멀티턴 해 생성시 옵션에 따른 기준 위치에 가까운 해 선택 <ul style="list-style-type: none"> <li>• 0 : posj(0,0,0,0,0,0) 조인트 각도 기준</li> <li>• 1 : 현재 조인트 각도 기준</li> </ul>

### Solution Space w.r.t. Robot Configuration

<b>Solution space</b>	<b>Binary</b>	<b>Shoulder</b>	<b>Elbow</b>	<b>Wrist</b>
0	000	Lefty	Below	No Flip
1	001	Lefty	Below	Flip
2	010	Lefty	Above	No Flip
3	011	Lefty	Above	Flip
4	100	Righty	Below	No Flip
5	101	Righty	Below	Flip
6	110	Righty	Above	No Flip
7	111	Righty	Above	Flip

### 리턴

값	설명
posj	Joint space point
status	<ul style="list-style-type: none"> <li>• 0 : 조인트 값 정상 반환</li> <li>• 1 : 동작 영역을 벗어남</li> <li>• 2 : 손목 축 특이점</li> </ul>

### 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시

예외	설명
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1  ## 2.x compatible code
2  x1 = posx(370.9, 719.7, 651.5, 90, -180, 0)
3  q1, status = ikin_norm(x1, 2, DR_BASE, ref_pos_opt=0) # 로봇끝단의 좌표가 x1이
   되는 관절각 q1 (8가지 경우 중 첫번째), posj(0,0,0,0,0,0) 조인트 각도 기준
4  # q1=posj(60.3, 81.0, -60.4, -0.0, 159.4, -29.7) (M1013, tcp=(0,0,0)경우)
5  movej(q1,v=10,a=20)
6
7  ## 3.2~ code
8  x1 = posx(370.9, 719.7, 651.5, 90, -180, 0, sol=2)
9  q1, status = ikin_norm(x1, ref=DR_BASE, ref_pos_opt=0) # Joint angle q1
   where the coordinate of the robot edge is x1 (second of 8 cases),
   reference joint position : posj(0,0,0,0,0,0)
10 # q1=posj(60.3, 81.0, -60.4, -0.0, 159.4, -29.7) (M1013, tcp=(0,0,0))
11 movej(q1,v=10,a=20)

```

## 관련 명령어

- set\_tcp(name)(p. 52)
- posj(J1=0, J2=0, J3=0, J4=0, J5=0, J6=0)(p. 29)
- posx(x=0, y=0, z=0, A=0, B=0, C=0, ori\_type=None, sol=None, turn=None)(p. 30)

## 3.2 모션 설정

### 3.2.1 set\_velj(vel)

#### 기능

본 명령어를 사용한 이후 조인트 모션(movej, movejx, amovej, amovejx)에서의 전역 속도를 설정합니다. 전역적으로 설정된 vel은 이후 movej() 호출 시 속도 인자를 명시적으로 입력하지 않는 경우에 default 속도로 적용됩니다.

## 인수

인수명	자료형	기본값	설명
vel	float	-	velocity(모든 축에 동일) 또는 velocity(축별 velocity)
	list (float[6])		

## 리턴

값	설명
0	성공

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시

## 예제

```

1 #1
2
3 Q1 = posj(0,0,90,0,90,0)
4
5 Q2 = posj(0,0,0,0,90,0)
6
7 movej(Q1, vel=10, acc=20)
8
9 set_velj(30) # 전역 조인트 속도를 30(deg/sec)로 설정하십시오.
10
11 set_accj(60) # 전역 조인트 가속도를 60(deg/sec2)로 설정하십시오. [set_accj() 참조]
12
13 movej(Q2) # Q2로의 조인트 모션 속도는 전역 속도인 30(deg/sec)입니다.
14
15 movej(Q1, vel=20, acc=40) # Q1으로의 조인트 모션 속도는 지정 속도인 20(deg/sec)입니다.
16
17 #2
18
19 set_velj(20.5) # 소수점 입력 가능합니다.
20
21 set_velj([10, 10, 20, 20, 30, 10]) # 축별 전역 속도 지정 가능합니다.

```

## 관련 명령어

- `set_accx(acc1, acc2)`(p. 50)
- (3.2.2-ko\_KR) `set_accx(acc)`<sup>1</sup>
- `movej()`(p. 57)
- `movejx()`(p. 67)
- `movesj()`(p. 77)
- `amovej()`(p. 100)
- `amovejx()`(p. 106)
- `amovesj()`(p. 113)

## 3.2.2 set\_accj(acc)

### 기능

본 명령어를 사용한 이후의 조인트 모션(movej, movejx, amovej, amovejx)에서의 전역 가속도를 설정합니다. 전역적으로 설정된 가속도는 이후 `movej()` 호출 시 가속도 인자를 명시적으로 입력하지 않는 경우에 default 가속도로 적용됩니다.

### 인수

인수명	자료형	기본값	설명
acc	float	-	acceleration(모든 축에 동일) 또는 acceleration(축별 acceleration)
	list (float[6])		

### 리턴

값	설명
0	성공

### 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시

<sup>1</sup> [https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/202179059/3.2.2-ko\\_KR+set\\_accx+acc](https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/202179059/3.2.2-ko_KR+set_accx+acc)

## 예제

```

1   #1
2
3   Q1 = posj(0,0,90,0,90,0)
4
5   Q2 = posj(0,0,0,0,90,0)
6
7   movej(Q1, vel=10, acc=20)
8
9   set_velj(30) # 전역 조인트 속도를 30(deg/sec)로 설정하십시오. #[set_velj() 참조]
10
11  set_accj(60) # 전역 조인트 가속도를 60(deg/sec2)로 설정하십시오.
12
13  movej(Q2) # Q2로의 조인트 모션 가속도는 전역 가속도인 60(deg/sec2)입니다.
14
15  movej(Q1, vel=20, acc=40) # Q1으로의 조인트 모션 가속도는 지정 가속도인 40(deg/sec2)
16  입니다.
17
18  #2
19
20  set_accj(30.55)
21  set_accj([30, 40, 30, 30, 30, 10])

```

## 관련 명령어

- [set\\_velj\(vel\)](#)(p. 45)
- [movej\(\)](#)(p. 57)
- [movejx\(\)](#)(p. 67)
- [movesj\(\)](#)(p. 77)
- [amovej\(\)](#)(p. 100)
- [amovejx\(\)](#)(p. 106)
- [amovesj\(\)](#)(p. 113)

### 3.2.3 set\_velx(vel1, vel2)

#### 기능

작업 공간 모션의 속도를 전역적으로 설정합니다. 전역적으로 설정된 속도 velx는 `movel()`, `amovel()`, `movec()`, `movesx()`과 같은 태스크 모션을 호출할 때 속도 값을 입력하지 않는 경우에는 default 속도로 적용됩니다.

설정값 중 vel1은 TCP의 선속도를, vel2는 TCP의 회전 속도를 정의합니다.

## 인수

인수명	자료형	기본값	설명
vel1	float	-	velocity 1
vel2	float	-	velocity 2

## 리턴

값	설명
0	성공

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시

## 예제

```

1 #1
2
3 P0 = posj(0,0,90,0,90,0)
4
5 movej(P0)
6
7 P1 = posx(400,500,800,0,180,0)
8
9 P2 = posx(400,500,500,0,180,0)
10
11 movel(P1, vel=10, acc=20)
12
13 set_velx(30,20)    # 전역 태스크 속도를 30(mm/sec), 20(deg/sec)로 설정하십시오.
14
15 set_accx(60,40)    # 전역 태스크 가속도를 60(mm/sec2), 40(deg/sec2)로 설정하십시오.
16
17 movel(P2)    # P2로의 태스크 모션 속도는 전역 속도인 30(mm/sec), 20(deg/sec)
18
19 movel(P1, vel=20, acc=40) # P1으로의 태스크 모션 속도는 지정 속도인 20(mm/sec),
20(deg/sec)입니다.
21 #2

```

```

22
23 set_velx(10.5, 19.4) # 소수점 입력 가능합니다.

```

## 관련 명령어

- [set\\_accx\(acc1, acc2\)](#)(p. 50)
- [move\(\)](#)(p. 62)
- [movec\(\)](#)(p. 71)
- [movesx\(\)](#)(p. 81)
- [moveb\(\)](#)(p. 85)
- [move\\_spiral\(\)](#)(p. 90)
- [amove\(\)](#)(p. 103)
- [amovec\(\)](#)(p. 109)
- [amovesx\(\)](#)(p. 116)
- [amoveb\(\)](#)(p. 119)
- [amove\\_spiral\(\)](#)(p. 124)

### 3.2.4 set\_accx(acc1, acc2)

#### 기능

작업 공간 모션의 가속도를 전역적으로 설정합니다. 전역적으로 설정된 가속도 accx는 move(), amove(), movec(), movesx()과 같은 태스크 모션을 호출할 때 가속도에 대한 값을 입력하지 않는 경우에는 default 가속도로 적용됩니다.

설정값 중 acc1은 TCP의 선가속도를, acc2는 TCP의 회전 가속도를 정의합니다.

#### 인수

인수명	자료형	기본값	설명
acc1	float	-	acceleration 1
acc2	float	-	acceleration 2

#### 리턴

값	설명
0	성공

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시

## 예제

```

1 P0 = posj(0,0,90,0,90,0)
2
3 movej(P0)
4
5 P1 = posx(400,500,800,0,180,0)
6
7 P2 = posx(400,500,500,0,180,0)
8
9 movel(P1, vel=10, acc=20)
10
11 set_velx(30,20) # 전역 태스크 속도를 30(mm/sec), 20(deg/sec)로 설정하십시오.
12
13 set_accx(60,40) # 전역 태스크 가속도를 60(mm/sec2), 40(deg/sec2)로 설정하십시오.
14
15 movel(P2)      # P2로의 태스크 모션 가속도는 전역 가속도인 60(mm/sec2), 40(deg/sec2)
16
17 movel(P1, vel=20, acc=40) # P1으로의 태스크 모션 가속도는 지정 가속도인 40(mm/sec),
                           # 40(deg/sec2)입니다.

```

## 관련 명령어

- [set\\_velx\(vel1, vel2\)\(p. 48\)](#)
- [movel\(\)\(p. 62\)](#)
- [movec\(\)\(p. 71\)](#)
- [movesx\(\)\(p. 81\)](#)
- [moveb\(\)\(p. 85\)](#)
- [move\\_spiral\(\)\(p. 90\)](#)
- [amovel\(\)\(p. 103\)](#)
- [amovec\(\)\(p. 109\)](#)
- [amovesx\(\)\(p. 116\)](#)
- [amoveb\(\)\(p. 119\)](#)
- [amove\\_spiral\(\)\(p. 124\)](#)

### 3.2.5 set\_tcp(name)

#### 기능

티치펜던트에 등록된 tcp의 이름을 호출하여 현재 tcp로 설정합니다.

#### 인수

인수명	자료형	기본값	설명
name	string	-	TP에 등록된 tcp 이름

#### 리턴

값	설명
0	성공
음수값	실패

#### 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

#### 예제

```

1 P0 = posj(0,0,90,0,90,0)
2
3 movej(P0)

```

```

4   set_tcp("tcp1") # TP에 tcp1으로 등록된 tcp 정보를 호출하여 현재의 tcp 값으로 설정하십시오.
5
6
7   P1 = posx(400,500,800,0,180,0)
8
9   movel(P1, vel=10, acc=20) # 인식한 tool 중심이 P1 위치로 이동

```

## 관련 명령어

- [fkin\(pos, ref, ori\\_type\)\(p. 37\)](#)
- [ikin\(pos, sol\\_space, ref, ref\\_pos\\_opt, iter\\_threshold\)\(p. 39\)](#)
- [movel\(\)\(p. 62\)](#)
- [movec\(\)\(p. 71\)](#)
- [movesx\(\)\(p. 81\)](#)
- [moveb\(\)\(p. 85\)](#)
- [move\\_spiral\(\)\(p. 90\)](#)
- [amovel\(\)\(p. 103\)](#)
- [amovec\(\)\(p. 109\)](#)
- [amovesx\(\)\(p. 116\)](#)
- [amoveb\(\)\(p. 119\)](#)
- [amove\\_spiral\(\)\(p. 124\)](#)

### 3.2.6 set\_ref\_coord(coord)

#### 기능

기준 좌표계를 설정합니다.

#### 인수

인수명	자료형	기본값	설명
coord	int	-	<p>기준 좌표</p> <ul style="list-style-type: none"> <li>DR_BASE: Base Coordinate</li> <li>DR_WORLD: World Coordinate</li> <li>DR_TOOL: Tool Coordinate</li> <li>user Coordinate: 사용자 정의</li> </ul>

#### 리턴

값	설명
0	성공

값	설명
음수값	실패

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 p0 = posj(0,0,90,0,90,0)
2
3 movej(p0, v=30, a=30)
4
5 x1 = posx(370.9, 419.7, 651.5, 90,-180,0)
6
7 movel(x1, v=100, a=100) # Base Coordinate 기준
8
9 uu1 = [-1, 1, 0] # 사용자좌표계의 x축방향벡터(Base Coordinate 기준)
10
11 vv1 = [1, 1, 0] # 사용자좌표계의 y축방향벡터(Base Coordinate 기준)
12
13 pos = posx(370.9, -419.7, 651.5, 0, 0, 0) # 사용자좌표계의 원점좌표
14
15 DR_USER1 = set_user_cart_coord(uu1, vv1, pos) # 사용자좌표계 설정
16
17 set_ref_coord(DR_USER1) # 사용자좌표계 DR_USER1를 전역좌표계로 지정
18
19 movel([0,0,0,0,0,0],v=100,a=100) # 별도의 ref지정 없는 경우 전역좌표를 따름
20
21     # DR_USER1좌표계의 원점 및 방향으로 이동
22
23 movel([0,200,0,0,0,0],v=100,a=100) # DR_USER1좌표계의 (0,200,0) 지점으로 이동

```

## 관련 명령어

- [moveJ\(\)](#)(p. 62)
- [moveJx\(\)](#)(p. 67)
- [moveC\(\)](#)(p. 71)
- [moveSx\(\)](#)(p. 81)
- [moveB\(\)](#)(p. 85)
- [move\\_spiral\(\)](#)(p. 90)
- [move\\_periodic\(\)](#)(p. 94)

### 3.2.7 set\_auto\_acceleration\_mode(mode, ratio)

#### 기능

사용자가 모션 입력 인자로 입력한 속도, 가속도 값으로 동작 시 허용 구동 토크를 벗어나는지 여부를 사전에 판단하여 동작 가능한 속도, 가속도 값으로 자동 조정 해주는 기능입니다. 입력 인자 mode는 자동 조정 기능 활성화 유무를 설정 합니다. 입력 인자 ratio는 자동 조정 비율을 설정합니다.

#### 알아두기

1. 가속도 자동 조정 기능과 연동되는 로봇 모델 : M 시리즈
2. 가속도 자동 조정 기능과 연동되는 모션 명령어 : moveJ, moveJx, moveL, moveC, moveB
3. 인자로 ratio를 기입하지 않을 시 초기값은 1.1로 택 타임 관점에서 10% 향상된 성능으로 속도, 가속도가 자동 조정 됩니다.(안정적으로 구동을 위한 권장 ratio는 1.0 입니다)
4. 각 로봇 모델별 허용 중량을 넘게 무게 설정 한 경우 가속도 자동 조정 기능이 자동으로 활성화 (DR\_ON, ratio=1.0)가 됩니다.

#### 주의

1. 가속도 자동 조정 기능 활성화 유무, 비율 조정 설정 값 변경은 해당 명령어를 통해서만 가능합니다. 프로그램 종료 시에도 해당 설정값(mode, ratio)은 유지 됩니다.
2. 각 로봇 모델별 허용 중량을 넘게 무게 설정 한 경우 해당 명령어를 통해 가속도 자동 조정 기능을 비활성화 하지 못하지만 설정 값(mode, ratio)은 유지 됩니다.
3. 경우에 따라 ratio를 1보다 크게 설정하여 택 타임 감소를 할 수 있지만 허용 구동 토크를 벗어나는 경우가 발생할 수 있습니다.
4. 블렌딩 구간을 지날 때 허용 구동 토크를 벗어나는 경우가 발생할 수 있습니다.

## 인수

인수명	자료형	기본값	설명
mode	int	DR_ON	<ul style="list-style-type: none"> <li>• DR_OFF(0) : 모드 비설정</li> <li>• DR_ON(1) : 모드 설정</li> </ul>
ratio	int	1.1	<ul style="list-style-type: none"> <li>• (0 ~ 1.2] 이내 설정 가능 하며 자동 조정 정도를 비율 만큼 조정</li> </ul>

## 리턴

값	설명
0	성공
음수값	실패

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 set_auto_acceleration_mode(DR_ON,1.0) # 기능 활성화, ratio=1.0
2
3 set_velj(60.0)
4 set_accj(100.0)
5 set_velx(250.0, 80.625)
6 set_accx(1000.0, 322.5)

```

```

7   home = posj(0,0,0,0,0,0)
8   p1 = posj(56.27, 42.77, 30.08, 0, 107.15, -56.27)
9   p2 = posx(501.69, 813.53, 51.47, 56.27, 180, -56.27)
10  p3 = posx(501.69, 813.53, 651.47, 56.27, -180, -56.27)
11
12
13 movej(home)
14 movej(p1,v=200, a=100)
15 movel(p2, v=1000, a=1000)
16 movel(p3, v=1000, a=1000)

```

## 관련 명령어

- movej()(p. 57)
- movejx()(p. 67)
- movel()(p. 62)
- movec()(p. 71)
- set\_tool(name, start\_time, transition\_time)(p. 190)

## 3.3 동기 모션

### 3.3.1 movej()

#### 기능

로봇이 현재 관절위치에서 목표 관절위치(pos)로 이동합니다.

- pos에 posj가 입력될 경우 기존처럼 movej로 동작
- pos에 posx가 입력될 경우 기존 movejx처럼 동작 (posx 입력할 경우 ref 변수 추가 입력 가능) → 2.x 버전의 movejx 기능과 동일합니다.

#### Note

- V3.2 버전부터, movejx 기능은 새로운 posx type과 함께 movej에서 지원됩니다.

#### 인수

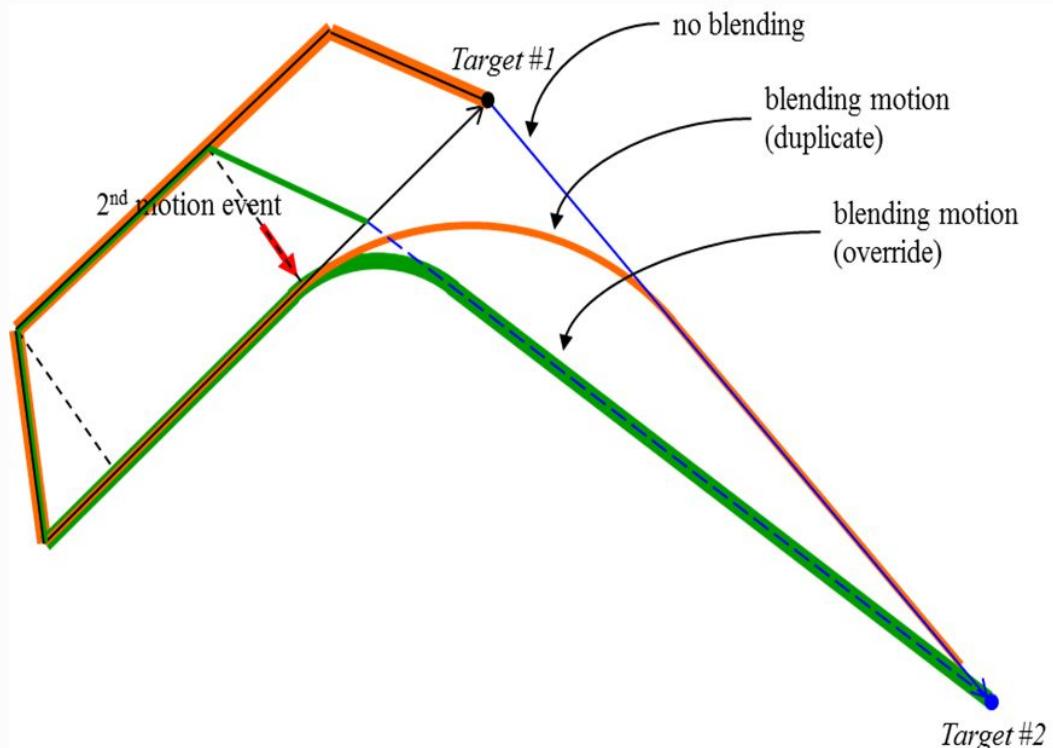
인수명	자료형	기본값	설명
pos	posj	-	posj 또는 joint angle list
	list (float[6])		posx

인수명	자료형	기본값	설명
	posx		
vel (v)	float	None	velocity(모든 축에 동일) 또는 velocity(축별 velocity)
	list (float[6])	None	
acc (a)	float	None	acceleration(모든 축에 동일) 또는 acceleration(축별 acceleration)
	list (float[6])	None	
time (t)	float	None	도달 시간 [sec]
radius (r)	float	None	blending 시 radius
mod	int	DR_MV_MOD_ABS	이동 기준 <ul style="list-style-type: none"> <li>• DR_MV_MOD_ABS : 절대</li> <li>• DR_MV_MOD_REL : 상대</li> </ul>
ra	int	DR_MV_RA_DUPLICATE	Reactive motion mode <ul style="list-style-type: none"> <li>• DR_MV_RA_DUPLICATE: duplicate</li> <li>• DR_MV_RA_OVERRIDE: override</li> </ul>
ref	int	None	reference coordinate <ul style="list-style-type: none"> <li>• DR_BASE: base coordinate</li> <li>• DR_WORLD: world coordinate</li> <li>• DR_TOOL: tool coordinate</li> <li>• user coordinate: 사용자 정의</li> </ul>

### ❶ 알아두기

- 단축 인수를 지원합니다. (v:vel, a:acc, t:time, r:radius)
- vel이 None인 경우, \_global\_velj이 적용됩니다. (\_global\_velj 초기값은 0.0이며, set\_velj에 의해 설정 가능)
- acc이 None인 경우, \_global\_accj이 적용됩니다. (\_global\_accj 초기값은 0.0이며, set\_accj에 의해 설정 가능)
- time 지정 시, vel, acc를 무시하고 time 기준으로 처리됩니다.
- time이 None인 경우, 0으로 처리됩니다.
- radius가 None인 경우, 블렌딩 구간인 경우는 blending radius로 처리되며 아닌 경우는 0으로 처리 됩니다.

- 진행 중인 모션(선행모션)이 종료되지 않은 상태에서 새로운 모션명령(후행모션)이 실행되면 선행 모션과 후행모션이 부드럽게 연결됩니다(모션 블렌딩 기능). 이 때 선행모션의 목표점을 유지하거나 취소할 수 있도록 결정하는 옵션 ra를 후행모션에 설정할 수 있습니다. (유지 :  $ra = DR\_MV\_RA\_DUPLICATE$  / 취소 :  $ra = DR\_MV\_RA\_ OVERRIDE$ ) 예를 들어 아래의 그림에서 목표점이 Target#1인 선행모션의 진행 중 '2nd motion event' 지점에서 후행모션이 실행된 경우 후행모션의 옵션  $ra = DR\_MV\_RA\_DUPLICATE$  이면 선행모션의 목표점을 유지하므로 주황색 궤적과 같이 움직이며  $ra = DR\_MV\_RA\_ OVERRIDE$  이면 선행모션의 목표점을 취소하므로 초록색 궤적과 같이 움직이게 됩니다.

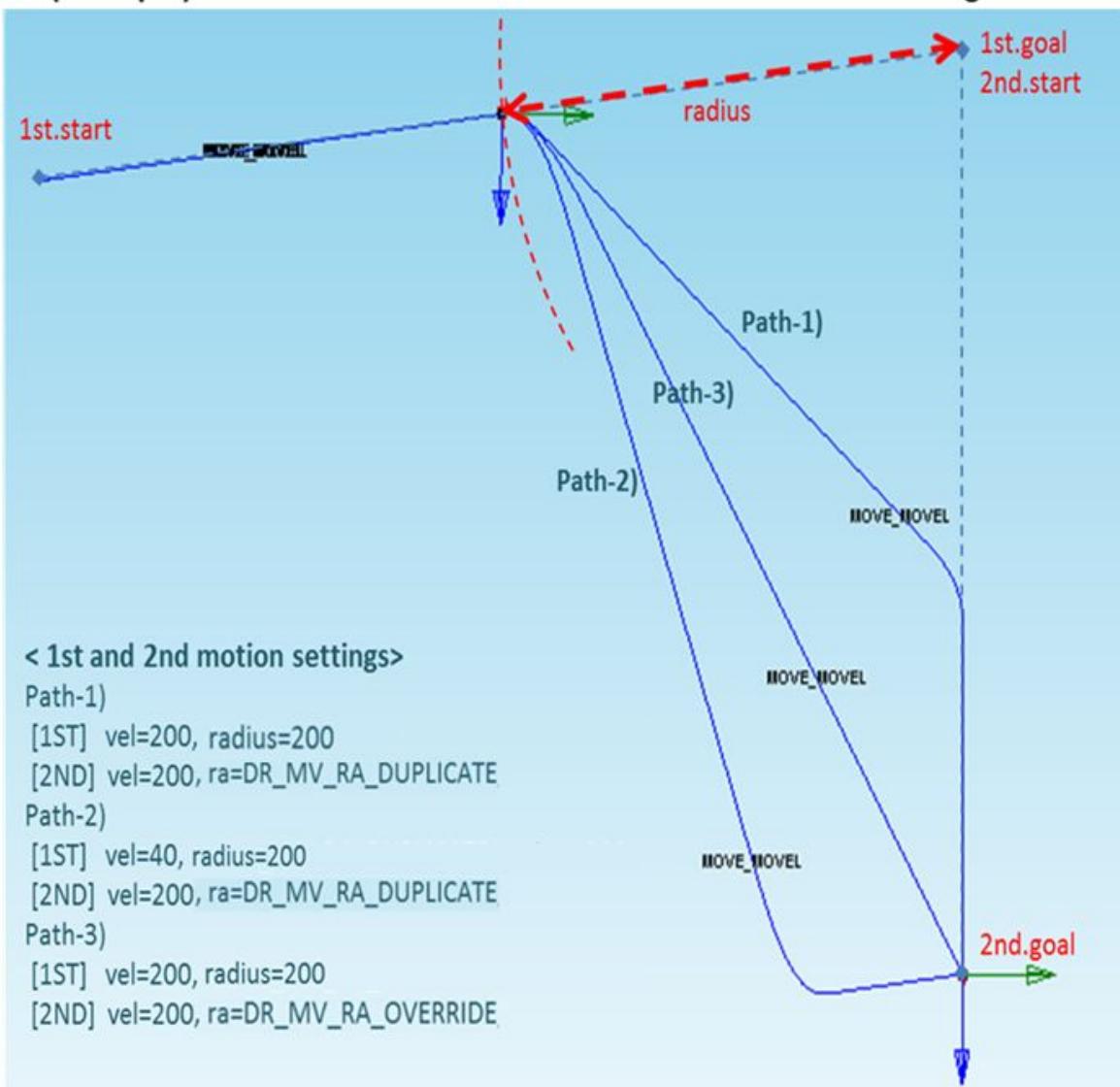


### 주의

$ra = DR\_MV\_RA\_DUPLICATE$  및  $radius > 0$  조건으로 후속 모션이 블렌딩 될 경우 선행모션의 잔여거리, 속도, 가속도로 결정되는 잔여모션시간이 후행모션의 모션시간보다 큰 경우 후행모션이 먼저 종료된 후 선행모션이 종료될 수 있습니다.

관련한 사항은 아래 이미지를 참고하십시오.

## &lt;(Example) Path differences accord. to 1st and 2nd motion settings&gt;



- SW V2.8 미만의 버전에서는 블렌딩 반지름 크기가 총 이동 거리의 1/2을 넘을 경우, 블렌딩 이후 모션에 영향을 주기 때문에 모션이 수행되지 않으며, 블렌딩 에러 발생 시 실행 중인 태스크 프로그램이 종료됩니다.
- SW V2.8 이상의 버전에서는 블렌딩 반지름 크기가 총 이동 거리의 1/2을 넘을 경우, 블렌딩 반지름 크기는 자동으로 총 이동 거리의 1/2로 변경되며, 변경 내역은 information log 메시지로 확인할 수 있습니다.

## 리턴

값	설명
0	성공

값	설명
음수값	실패

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 Q1 = posj(0,0,90,0,90,0)
2
3 Q2 = posj(0,0,0,0,90,0)
4
5 movej(Q1, vel=10, acc=20)
6
7 # 속도 10(deg/sec), 가속도 20(deg/sec2)로 Q1관절각으로 이동
8
9 movej(Q2, time=5)
10
11 # Q2관절각까지 5초의 도착시간을 가지고 이동
12
13 movej(Q1, v=30, a=60, r=200)
14
15 # Q1관절각으로 이동하며 Q1의 공간위치로부터 200mm의 거리가 될 때 다음
16
17 # 모션을 수행하도록 설정
18
19 movej(Q2, v=30, a=60, ra= DR_MV_RA_OVERRIDE)
20
21 # 직전모션을 즉시 종료시키며 Blending하여 Q2관절각으로 이동
22
23 ## movejx function
24 P1 = posx(400,500,800,0,180,0, sol=2)

```

25 movej (P1, v=30, a=60)

## 관련 명령어

- posj(J1=0, J2=0, J3=0, J4=0, J5=0, J6=0)(p. 29)
- set\_velj(vel)(p. 45)
- set\_accj(acc)(p. 47)
- amovej()(p. 100)

### 3.3.2 movel()

#### 기능

로봇이 작업 공간 안에서 목표 위치(pos)로 직선을 따라 이동합니다.

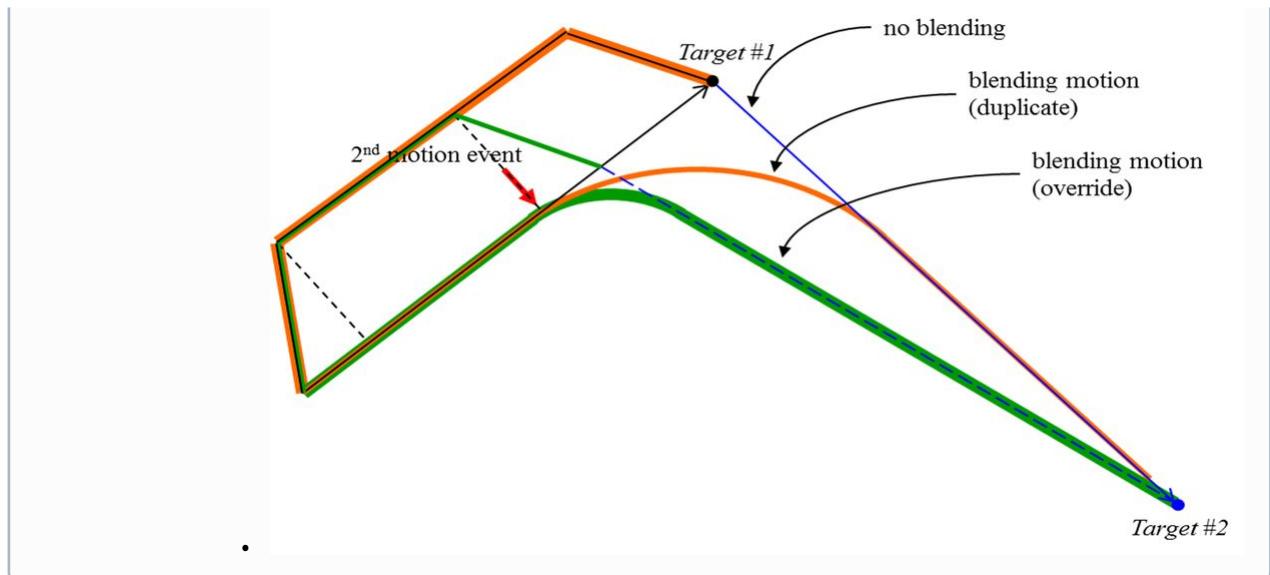
#### 인수

인수명	자료형	기본값	설명
pos	posx	-	posx 또는 position list
	list (float[6])		
vel (v)	float	None	velocity 또는 velocity1, velocity2
	list (float[2])	None	
acc (a)	float	None	acceleration 또는 acceleration1, acceleration2
	list (float[2])	None	
time (t)	float	None	도달 시간 [sec] <ul style="list-style-type: none"> <li>• time 지정 시, vel, acc를 무시하고 time 기준으로 처리</li> </ul>
radius (r)	float	None	blending 시 radius
ref	int	None	reference coordinate <ul style="list-style-type: none"> <li>• DR_BASE: base coordinate</li> <li>• DR_WORLD: world coordinate</li> <li>• DR_TOOL: tool coordinate</li> <li>• user coordinate: 사용자 정의</li> </ul>

인수명	자료형	기본값	설명
mod	int	DR_MV_MOD_ABS	<p>이동 기준</p> <ul style="list-style-type: none"> <li>DR_MV_MOD_ABS : 절대</li> <li>DR_MV_MOD_REL : 상대</li> </ul>
ra	int	DR_MV_RA_DUPLICATE	<p>Reactive motion mode</p> <ul style="list-style-type: none"> <li>DR_MV_RA_DUPLICATE: duplicate</li> <li>DR_MV_RA_OVERRIDE: override</li> </ul>
app_type	int	DR_MV_APP_NONE	<p>어플리케이션 연동 옵션</p> <ul style="list-style-type: none"> <li>DR_MV_APP_NONE: 연동없음</li> <li>DR_MV_APP_WELD: 용접연동</li> </ul>

### ❶ 알아두기

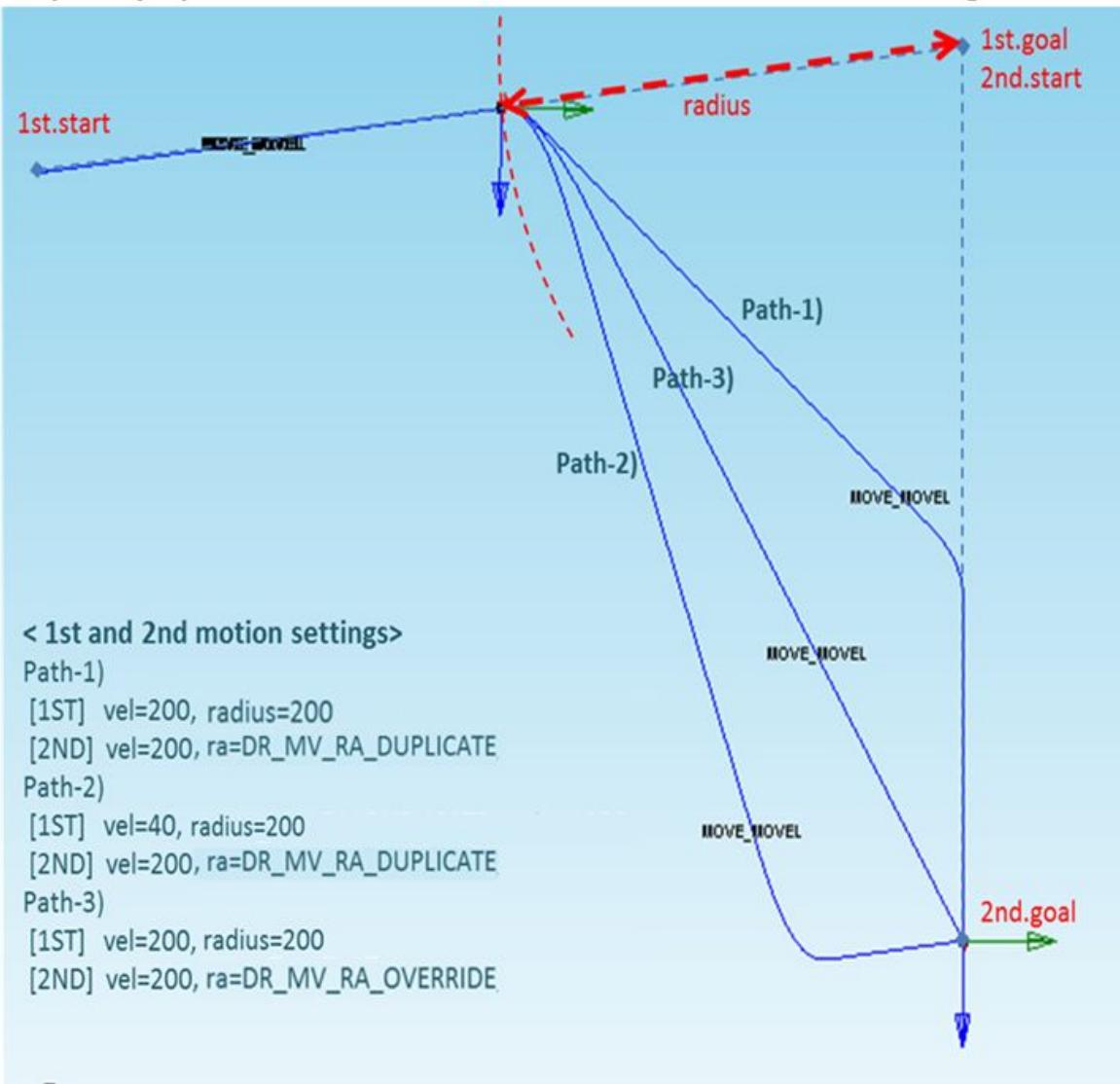
- 단축 인수를 지원합니다. (v:vel, a:acc, t:time, r:radius)
- vel가 None인 경우, \_global\_velx이 적용됩니다. (\_global\_velx 초기값은 0.0이며, set\_velx에 의해 설정 가능)
- acc가 None인 경우, \_global\_accx이 적용됩니다. (\_global\_accx 초기값은 0.0이며, set\_accx에 의해 설정 가능)
- vel에 하나의 인자를 입력한 경우(예를 들어, vel=30) 입력된 인자는 모션의 선속도에 대응되며, 각속도는 선속도에 비례하여 결정됩니다.
- acc에 하나의 인자를 입력한 경우(예를 들어, acc=60) 입력된 인자는 모션의 선가속도에 대응되며, 각가속도는 선가속도에 비례하여 결정됩니다.
- time 지정 시, vel, acc를 무시하고 time 기준으로 처리됩니다.
- time이 None인 경우, 0으로 처리됩니다.
- radius가 None이고 블렌딩 구간인 경우는 blending radius로 처리되고 아닌 경우는 0으로 처리됩니다.
- ref가 None인 경우에는 \_g\_coord이 적용됩니다. (\_g\_coord 초기값은 DR\_BASE이며, set\_ref\_coord 명령에 의해 설정 가능)
- app\_type이 DR\_MV\_APP\_WELD인 경우에 이동속도는 vel인자의 입력값이 아닌 app\_weld\_set\_weld\_cond()에 입력된 속도설정으로 대체됩니다.
- 진행 중인 모션(선행모션)이 종료되지 않은 상태에서 새로운 모션명령(후행모션)이 실행되면 선행 모션과 후행모션이 부드럽게 연결됩니다(모션 블렌딩 기능). 이 때 선행모션의 목표점을 유지하거나 취소할 수 있도록 결정하는 옵션 ra를 후행모션에 설정할 수 있습니다. (유지 : ra=DR\_MV\_RA\_DUPLICATE / 취소 : ra=DR\_MV\_RA\_OVERRIDE) 예를 들어 아래의 그림에서 목표점이 Target#1인 선행모션의 진행 중 '2<sup>nd</sup> motion event' 지점에서 후행모션이 실행된 경우 후행모션의 옵션 ra=DR\_MV\_RA\_DUPLICATE 이면 선행모션의 목표점을 유지하므로 주황색 궤적과 같이 움직이며 ra=DR\_MV\_RA\_OVERRIDE 이면 선행모션의 목표점을 취소하므로 초록색 궤적과 같이 움직이게 됩니다.



### ⚠ 주의

ra=DR\_MV\_RA\_DUPLICATE 및 radius>0 조건으로 후속 모션이 블렌딩 될 경우 선행모션의 잔여거리, 속도, 가속도로 결정되는 잔여모션시간이 후행모션의 모션시간보다 큰 경우 후행모션이 먼저 종료된 후 선행모션이 종료될 수 있습니다. 관련한 사항은 아래 이미지를 참고하십시오.

## &lt;(Example) Path differences accord. to 1st and 2nd motion settings&gt;



- SW V2.8 미만의 버전에서는 블렌딩 반지름 크기가 총 이동 거리의 1/2을 넘을 경우, 블렌딩 이후 모션에 영향을 주기 때문에 모션이 수행되지 않으며, 블렌딩 에러 발생 시 실행 중인 태스크 프로그램이 종료됩니다.
- SW V2.8 이상의 버전에서는 블렌딩 반지름 크기가 총 이동 거리의 1/2을 넘을 경우, 블렌딩 반지름 크기는 자동으로 총 이동 거리의 1/2로 변경되며, 변경 내역은 information log 메시지로 확인할 수 있습니다.

## 리턴

값	설명
0	성공

값	설명
음수값	실패

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 P0 = posj(0,0,90,0,90,0)
2
3 movej(P0, v=30, a=30)
4
5 P1 = posx(400,500,800,0,180,0)
6
7 P2 = posx(400,500,500,0,180,0)
8
9 P3 = posx(30,30,30,0,0,0)
10
11 movel(P1, vel=30, acc=100)
12
13 # 속도 30(mm/sec), 가속도 100(mm/sec2)로 P1위치로 이동
14
15 movel(P2, time=5)
16
17 # P2위치로 5초의 도착시간을 가지고 이동
18
19 movel(P3, time=5, ref=DR_TOOL, mod=DR_MV_MOD_REL)
20
21 # 시작위치에서 Tool좌표계기준으로 P3만큼의 상대위치로 5초의 도착시간을
22
23 # 가지고 이동시킴
24

```

```

25   movel(P2, time=5, r=10)
26
27   # P2위치까지 5초의 도착시간을 가지고 이동시키며 P2 위치로부터 10mm의
28
29   # 거리가 될 때 다음 모션을 수행하도록 설정

```

## 관련 명령어

- posx(x=0, y=0, z=0, A=0, B=0, C=0, ori\_type=None, sol=None, turn=None)(p. 30)
- set\_velx(vel1, vel2)(p. 48)
- (3.2.2-ko\_KR) set\_velx(vel)<sup>2</sup>
- set\_accx(acc1, acc2)(p. 50)
- (3.2.2-ko\_KR) set\_accx(acc)<sup>3</sup>
- set\_tcp(name)(p. 52)
- set\_ref\_coord(coord)(p. 53)
- amovel()(p. 103)

### 3.3.3 movejx()

#### 기능

로봇이 관절 공간 안에서 목표 위치(pos)로 이동합니다.

목표 위치는 작업공간 상의 posx형으로 입력하므로 movel과 동일하게 이동합니다. 하지만 이 로봇의 모션은 관절공간에서 이루어지기 때문에 목표 위치까지 직선경로가 보장되지 않습니다. 추가적으로 하나의 작업공간좌표(posx)에 대응하는 8가지의 관절조합형태(robot configuration)중 하나를 sol(solution space)에 지정하여야 합니다.

#### **Note**

- V3.2 버전부터, movejx는 2.x 호환 형태로만 제공되고, 새로운 posx type에 대한 해당 기능은 movej에서 지원됩니다.

#### 인수

인수명	자료형	기본값	설명
pos	posx	-	posx 또는 position list
	list (float[6])		

<sup>2</sup> [https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/202473924/3.2.2-ko\\_KR+set\\_velx+vel](https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/202473924/3.2.2-ko_KR+set_velx+vel)

<sup>3</sup> [https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/202179059/3.2.2-ko\\_KR+set\\_accx+acc](https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/202179059/3.2.2-ko_KR+set_accx+acc)

인수명	자료형	기본값	설명
vel (v)	float	None	velocity(모든 축에 동일) 또는 velocity(축별 velocity)
	list (float[6])	None	
acc (a)	float	None	acceleration(모든 축에 동일) 또는 acceleration(축별 acceleration)
	list (float[6])	None	
time (t)	float	None	도달 시간 [sec]
radius (r)	float	None	blending 시 radius
ref	int	None	reference coordinate <ul style="list-style-type: none"> <li>DR_BASE: base coordinate</li> <li>DR_WORLD: world coordinate</li> <li>DR_TOOL: tool coordinate</li> <li>user coordinate: 사용자 정의</li> </ul>
mod	int	DR_MV_MOD_ABS	이동 기준 <ul style="list-style-type: none"> <li>DR_MV_MOD_ABS: 절대</li> <li>DR_MV_MOD_REL: 상대</li> </ul>
ra	int	DR_MV_RA_DUPLICATE	Reactive motion mode <ul style="list-style-type: none"> <li>DR_MV_RA_DUPLICATE: duplicate</li> <li>DR_MV_RA_OVERRIDE: override</li> </ul>
sol	int	0	Solution space

### ❶ 알아두기

- 단축 인수를 지원합니다. (v:vel, a:acc, t:time, r:radius)
- vel이 None인 경우, \_global\_velj가 적용됩니다. (\_global\_velj 초기값은 0.0이며, set\_velj에 의해 설정 가능)
- acc가 None인 경우, \_global\_accj가 적용됩니다. (\_global\_accj 초기값은 0.0이며, set\_accj에 의해 설정 가능)
- time을 지정할 경우 vel, acc를 무시하고 time 기준으로 처리됩니다.
- time이 None인 경우, 0으로 처리됩니다.
- radius가 None이고 블렌딩 구간인 경우는 blending radius로 처리되며 아닌 경우는 0으로 처리됩니다.

- ref가 None인 경우 \_g\_coord가 적용됩니다. (\_g\_coord 초기값은 DR\_BASE이며, set\_ref\_coord 명령에 의해 설정 가능)
- 상대모션으로 입력하는 경우(mod=DR\_MV\_MOD\_REL), 선행모션에 블렌딩을 사용하는 경우 에러가 발생하므로 movej() 또는 movel()을 이용하여 블렌딩하는 것을 권장합니다.
- 옵션 ra 및 vel/acc에 따른 블렌딩을 수행할 경우 movej(), movel() 설명을 참조하십시오.

### 주의

- SW V2.8 미만의 버전에서는 블렌딩 반지름 크기가 총 이동 거리의 1/2을 넘을 경우, 블렌딩 이후 모션에 영향을 주기 때문에 모션이 수행되지 않으며, 블렌딩 에러 발생 시 실행 중인 태스크 프로그램이 종료됩니다.
- SW V2.8 이상의 버전에서는 블렌딩 반지름 크기가 총 이동 거리의 1/2을 넘을 경우, 블렌딩 반지름 크기는 자동으로 총 이동 거리의 1/2로 변경되며, 변경 내역은 information log 메시지로 확인할 수 있습니다.

### Robot configuration (형태 solution space)

Solution space	Binary	Shoulder	Elbow	Wrist
0	000	Lefty	Below	No Flip
1	001	Lefty	Below	Flip
2	010	Lefty	Above	No Flip
3	011	Lefty	Above	Flip
4	100	Righty	Below	No Flip
5	101	Righty	Below	Flip
6	110	Righty	Above	No Flip
7	111	Righty	Above	Flip

### 리턴

값	설명
0	성공
음수값	오류

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 P0 = posj(0,0,90,0,90,0)
2
3 movej(P0, v=30, a=30)
4
5 P1 = posx(400,500,800,0,180,0)
6
7 P2 = posx(400,500,500,0,180,0)
8
9 movel(P2, vel=100, acc=200)      # P2로 직선이동
10
11 X_tmp, sol_init = get_current_posx()    # P2위치에서 현재의 solution space를 얻어
12 옴
13
14 movejx(P1, vel=30, acc=60, sol=sol_init)
15 # (가) 속도 30(deg/sec), 60(deg/sec2)로 TCP끝단이 P1위치일 때의 관절각으로
16
17 # 이동 (직전P2위치에서의 solution space유지)
18
19 movejx(P2, time=5, sol=2)
20
21 # TCP끝단이 P2위치일 때의 관절각으로 5초의 도착시간을 가지고
22
23 # 이동 (solution space를 강제로 2로 지정)
24
25 movejx(P1, vel=[10, 20, 30, 40, 50, 60], acc=[20, 20, 30, 30, 40, 40],
26 radius=100, sol=2)
27 # TCP끝단이 P1위치일 때의 관절각으로 이동시키며 P1 위치로부터 100mm의

```

```

28
29   # 거리가 될 때 다음 모션을 수행하도록 설정
30
31   movejx(P2, v=30, a=60, ra= DR_MV_RA_OVERRIDE, sol=2)
32
33   # 직전모션을 즉시 종료시키며 Blending하여 TCP끝단이 P2위치일 때의
34
35   # 관절각으로 이동

```

## 관련 명령어

- [posx\(x=0, y=0, z=0, A=0, B=0, C=0, ori\\_type=None, sol=None, turn=None\)\(p. 30\)](#)
- [set\\_velj\(vel\)\(p. 45\)](#)
- [set\\_accj\(acc\)\(p. 47\)](#)
- [get\\_current\\_posx\(ref, ori\\_type\)\(p. 168\)](#)
- [amovejx\(\)\(p. 106\)](#)

### 3.3.4 movec()

#### 기능

작업공간(task space)을 기준으로 로봇이 현재 위치에서 경유점(pos1)를 지나 목표위치(pos2)까지의 원호 또는 지정한 각도까지 원호를 따라 이동합니다.

#### 인수

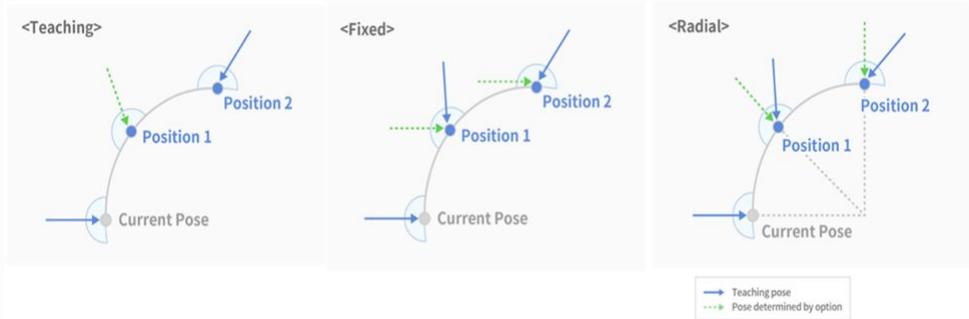
인수명	자료형	기본값	설명
pos	posj	-	posx 또는 position list
	list (float[6])		
pos2	posx		posx 또는 position list
	list (float[6])		
vel (v)	float	None	velocity 또는 velocity1, velocity2
	list (float[2])	None	
acc (a)	float	None	acceleration 또는 acceleration1, acceleration2
	list (float[2])	None	

인수명	자료형	기본값	설명
time (t)	float	None	도달 시간 [sec]
radius (r)	float	None	blending시 radius
ref	int	None	reference coordinate <ul style="list-style-type: none"> <li>DR_BASE: base coordinate</li> <li>DR_WORLD: world coordinate</li> <li>DR_TOOL: tool coordinate</li> <li>user coordinate: 사용자 정의</li> </ul>
mod	int	DR_MV_MOD_ABS	이동 기준 <ul style="list-style-type: none"> <li>DR_MV_MOD_ABS: 절대</li> <li>DR_MV_MOD_REL: 상대</li> </ul>
angle (an)	float	None	angle 또는 angle1, angle2
list (float[2])			
ra	int	DR_MV_RA_DUPLICATE	Reactive motion mode <ul style="list-style-type: none"> <li>DR_MV_RA_DUPLICATE: duplicate</li> <li>DR_MV_RA_OVERRIDE: override</li> </ul>
ori	int	DR_MV_ORI_TEACH	이동자세 선택 <ul style="list-style-type: none"> <li>DR_MV_ORI_TEACH: 교시자세</li> <li>DR_MV_ORI_FIXED: 고정자세</li> <li>DR_MV_ORI_RADIAL: 원주구속 자세</li> </ul>
app_type	int	DR_MV_APP_NONE	어플리케이션 연동 옵션 <ul style="list-style-type: none"> <li>DR_MV_APP_NONE: 연동없음</li> <li>DR_MV_APP_WELD: 용접연동</li> </ul>

### ❶ 알아두기

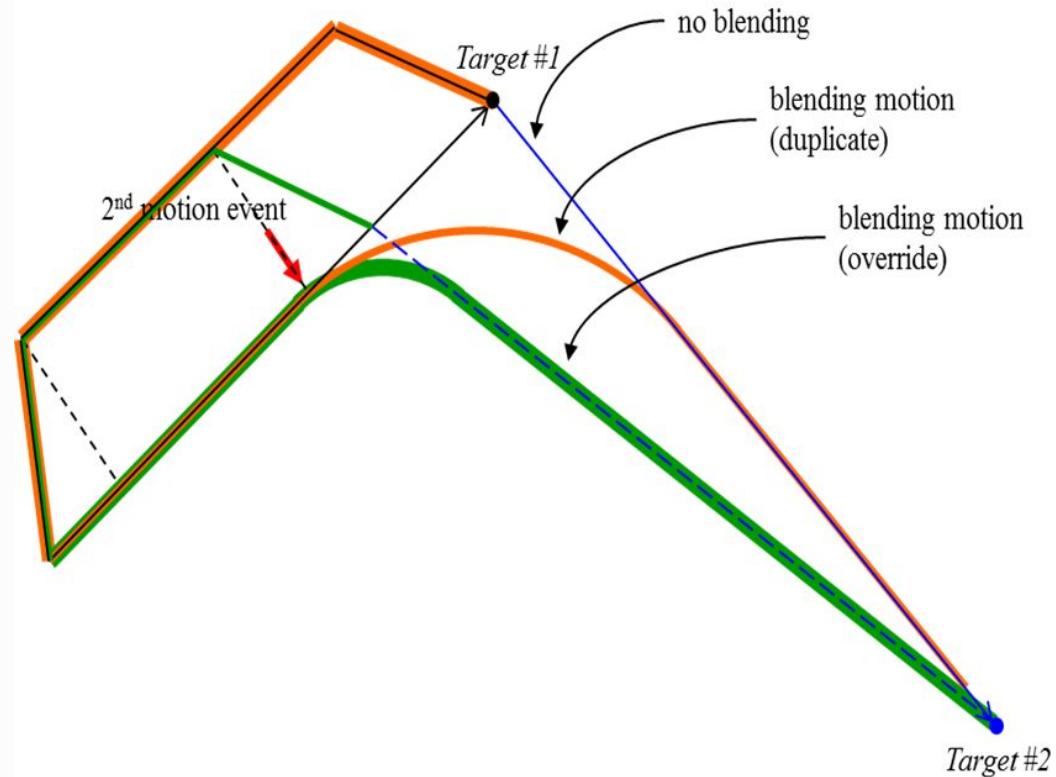
- 단축 인수를 지원합니다. (v:vel, a:acc, t:time, r:radius, angle:an)
- vel이 None인 경우 \_global\_velx가 적용됩니다. (\_global\_velx 초기값은 0.0이며, set\_velx에 의해 설정 가능)
- acc가 None인 경우 \_global\_accx가 적용됩니다. (\_global\_accx 초기값은 0.0이며, set\_accx에 의해 설정 가능)

- vel에 하나의 인자를 입력한 경우(예를 들어, vel=30) 입력된 인자는 모션의 선속도에 대응되며, 각속도는 선속도에 비례하여 결정됩니다.
- acc에 하나의 인자를 입력한 경우(예를 들어, acc=60) 입력된 인자는 모션의 선가속도에 대응되며, 각가속도는 선가속도에 비례하여 결정됩니다.
- time을 지정할 경우 vel, acc를 무시하고 time 기준으로 처리됩니다.
- time이 None인 경우 0으로 처리됩니다.
- radius가 None이고 블렌딩 구간인 경우는 blending radius로 처리되며 아닌 경우는 0으로 처리됩니다.
- ref가 None인 경우 \_g\_coord가 적용됩니다. (\_g\_coord 초기값은 DR\_BASE이며, set\_ref\_coord 명령에 의해 설정 가능)
- mod가 DR\_MV\_MOD\_REL인 경우 pos1과 pos2는 각각 앞선 pos에 대한 상대좌표로 정의됩니다. (pos1은 시작점 대비 상대좌표, pos2는 pos1 대비 상대좌표)
- angle이 None일 경우 0으로 처리됩니다.
- angle이 한 개만 입력된 경우 angle은 Circular path 상의 총 회전각이 적용됩니다.
- angle이 두 개가 입력된 경우, angle1은 circular path 상에서 정속으로 이동하는 총 회전각을, angle2는 가속과 감속을 위한 회전 구간의 회전각을 의미합니다. 이때 총 이동각은 angle1 + 2 X angle2만큼 circular path 상을 움직입니다.
- app\_type이 DR\_MV\_APP\_WELD인 경우에 이동속도는 vel인자의 입력값이 아닌 app\_weld\_set\_weld\_cond()에 입력된 속도설정으로 대체됩니다.
- ori(이동자세 선택)옵션은 아래와 같이 정의됩니다.
  - a. DR\_MV\_ORI\_TEACH(교시자세) : 현재 자세로부터 포즈 2의 교시 자세까지 이동거리에 비례하여 자세를 변화하며 이동합니다. 포즈1의 교시자세는 무시됩니다.
  - b. DR\_MV\_ORI\_FIXED(고정자세) : 현재 자세를 포즈2 지점까지 유지한 채 경로를 따라 이동합니다.
  - c. DR\_MV\_ORI\_RADIAL(원주구속자세) : 현재 자세에서의 반경방향의 자세를 포즈2 지점까지 유지한 채 이동합니다.



- 진행 중인 모션(선행모션)이 종료되지 않은 상태에서 새로운 모션명령(후행모션)이 실행되면 선행 모션과 후행모션이 부드럽게 연결됩니다(모션 블렌딩 기능). 이 때 선행모션의 목표점을 유지하거나 취소할 수 있도록 결정하는 옵션 ra를 후행모션에 설정할 수 있습니다. (유지 : ra=DR\_MV\_RA\_DUPLICATE / 취소 : ra=DR\_MV\_RA\_OVERRIDE) 예를 들어 아래의 그림에서 목표점이 Target#1인 선행모션의 진행 중 '2nd motion event' 지점에서 후행모션이 실행된 경우 후행모션의 옵션 ra=DR\_MV\_RA\_DUPLICATE 이면 선행모션의 목표점을 유지하므로 주황색 궤적과 같이 움직이며 ra=DR\_MV\_RA\_OVERRIDE 이면 선행모션의 목표점을 취소하므로 초록색 궤적과 같이 움직이

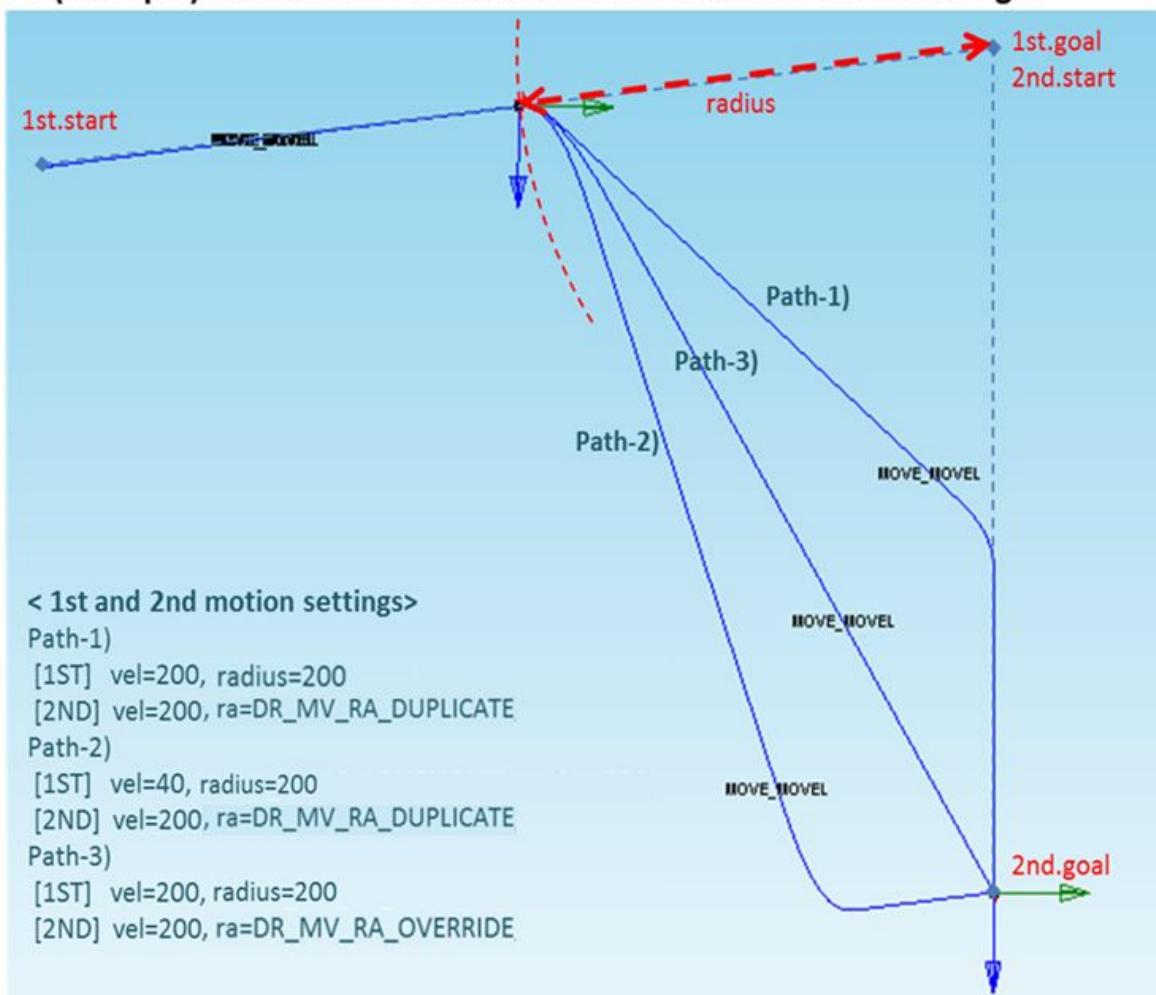
게 됩니다.



#### ⚠ 주의

ra=DR\_MV\_RA\_DUPLICATE 및 radius>0 조건으로 후속 모션이 블렌딩 될 경우 선행모션의 잔여거리, 속도, 가속도로 결정되는 잔여모션시간이 후행모션의 모션시간보다 큰 경우 후행모션이 먼저 종료된 후 선행모션이 종료될 수 있습니다. 관련한 사항은 아래 이미지를 참고하십시오.

## &lt; (Example) Path differences accord. to 1st and 2nd motion settings&gt;



- SW V2.8 미만의 버전에서는 블렌딩 반지름 크기가 총 이동 거리의 1/2을 넘을 경우, 블렌딩 이후 모션에 영향을 주기 때문에 모션이 수행되지 않으며, 블렌딩 에러 발생 시 실행 중인 태스크 프로그램이 종료됩니다.
- SW V2.8 이상의 버전에서는 블렌딩 반지름 크기가 총 이동 거리의 1/2을 넘을 경우, 블렌딩 반지름 크기는 자동으로 총 이동 거리의 1/2로 변경되며, 변경 내역은 information log 메시지로 확인할 수 있습니다.

## 리턴

값	설명
0	성공
음수값	오류

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 #1
2
3 P0 = posj(0,0,90,0,90,0)
4
5 movej(P0)
6
7 set_velx(30,20) # 전역 태스크 속도를 30(mm/sec), 20(deg/sec)로 설정
8
9 set_accx(60,40) # 전역 태스크 가속도를 60(mm/sec2), 40(deg/sec2)로 설정
10
11
12 P1 = posx(400,500,800,0,180,0)
13
14 P2 = posx(400,500,500,0,180,0)
15
16 P3 = posx(100, 300, 700, 45, 0, 0)
17
18 P4 = posx(500, 400, 800, 45, 45, 0)
19
20
21
22 movec(P1, P2, vel=30)
23
24 # 속도 30(mm/sec), 전역가속도 60(mm/sec2)로 P1을 경유하여 P2에 이르는
25
26 # 원호궤적을 따라 이동
27
28 movej(P0)
29

```

```

30
31 movec(P3, P4, vel=30, acc=60)
32
33 # 속도 30(mm/sec), 가속도 60(mm/sec2)로 P3를 경유하여 P4에 이르는
34
35 # 원호궤적을 따라 이동
36
37 movej(P0)
38
39 movec(P2, P1, time=5)
40
41 # 전역(가)속도 30(mm/sec), 60(mm/sec2)로 P2를 경유하여 시점 5초에
42
43 # P1에 이르는 원호궤적을 이동
44
45 movec(P3, P4, time=3, radius=100)
46
47 # P3를 경유하여 P4로 이동하는 원호궤적을 3초의 도착시간을 가지고
48
49 # 이동시키며 P4 위치로부터 100mm의 거리가 될 때 다음 모션을 수행하도록
50
51 # 설정
52
53 movec(P2, P1, ra=DR_MV_RA_OVERRIDE)
54
55 # 직전모션을 즉시 종료시키며 Blending하여 P1위치로 이동

```

## 관련 명령어

- posx(x=0, y=0, z=0, A=0, B=0, C=0, ori\_type=None, sol=None, turn=None)(p. 30)
- set\_velx(vel1, vel2)(p. 48)
- (3.2.2-ko\_KR) set\_velx(vel)<sup>4</sup>
- set\_accx(acc1, acc2)(p. 50)
- (3.2.2-ko\_KR) set\_accx(acc)<sup>5</sup>
- set\_tcp(name)(p. 52)
- set\_ref\_coord(coord)(p. 53)
- amovec()(p. 109)

## 3.3.5 movesj()

### 기능

현재 위치에서 pos\_list로 입력된 관절공간(joint space)의 경유점들을 거쳐 목표위치(pos\_list의 마지막 경유점)까지 연결되는 스플라인 곡선경로를 따라 이동합니다.

<sup>4</sup> [https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/202473924/3.2.2-ko\\_KR+set\\_velx+vel](https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/202473924/3.2.2-ko_KR+set_velx+vel)

<sup>5</sup> [https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/202179059/3.2.2-ko\\_KR+set\\_accx+acc](https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/202179059/3.2.2-ko_KR+set_accx+acc)

입력된 속도/가속도는 경로 중 최대 속도/가속도를 의미하며 입력되는 경유점의 위치에 따라 모션 중의 감속, 가속이 결정됩니다.

## 인수

인수명	자료형	기본값	설명
pos_list	list (posj)	-	posj list
vel (v)	float	None	velocity(모든 축에 동일) 또는 velocity(축별 velocity)
	list (float[6])		
acc (a)	float	None	acceleration(모든 축에 동일) 또는 acceleration(축별 acceleration)
	list (float[6])		
time (t)	float	None	도달 시간 [sec]
mod	int	DR_MV_MOD_ABS	이동 기준 <ul style="list-style-type: none"> <li>• DR_MV_MOD_ABS : 절대</li> <li>• DR_MV_MOD_REL : 상대</li> </ul>

### ❶ 알아두기

- 단축 인수를 지원합니다. (v:vel, a:acc, t:time)
- vel이 None인 경우 \_global\_velj가 적용됩니다. (\_global\_velj 초기값은 0.0이며, set\_velj에 의해 설정 가능)
- acc이 None인 경우 \_global\_accj가 적용됩니다. (\_global\_accj 초기값은 0.0이며, set\_accj에 의해 설정 가능)
- time을 지정할 경우 vel, acc를 무시하고 time 기준으로 처리됩니다.
- time이 None인 경우 0으로 처리됩니다.
- mod가 DR\_MV\_MOD\_REL인 경우 pos\_list의 각 pos는 앞 선 pos에 대한 상대좌표로 정의됩니다. (pos\_list=[q1, q2, ..., q(n-1), q(n)]로 이루어질 때 q1은 시작점 대비 상대각도, q(n)은 q(n-1) 대비 상대좌표)
- 선행모션과 후행모션에 대한 온라인 블렌딩은 지원하지 않습니다.

## 리턴

값	설명
0	성공
음수값	오류

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 #CASE 1) 절대각도 입력 (mod= DR_MV_MOD_ABS)
2
3 q0 = posj(0,0,0,0,0,0)
4
5 movej(q0, vel=30, acc=60) # 초기위치(q0)로 joint모션 이동
6
7 q1 = posj(10, -10, 20, -30, 10, 20) # posj 변수(관절각) q1 정의
8
9 q2 = posj(25, 0, 10, -50, 20, 40)
10
11 q3 = posj(50, 50, 50, 50, 50, 50)
12
13 q4 = posj(30, 10, 30, -20, 10, 60)
14
15 q5 = posj(20, 20, 40, 20, 0, 90)
16
17
18

```

```

19 qlist = [q1, q2, q3, q4, q5] # q1~q5를 경유점 집합으로 하는 리스트(qlist) 정의
20
21
22
23 movesj(qlist, vel=30, acc=100)
24 # qlist에 정의된 경유점 집합을 연결하는 스플라인 곡선을 최대속도
25 # 30(mm/sec), 최대가속도 100(mm/sec2)로 움직임
26
27
28
29
30
31 #CASE 2) 상대각도 입력 (mod= DR_MV_MOD_REL)
32
33 q0 = posj(0,0,0,0,0,0)
34
35 movej(q0, vel=30, acc=60) # 초기위치(q0)로 joint모션 이동
36
37 dq1 = posj(10, -10, 20, -30, 10, 20) # q0에 대한 상대관절각 dq1 정의
38 (q1=q0+dq1)
39 dq2 = posj(15, 10, -10, -20, 10, 20) # q1에 대한 상대관절각 dq2 정의
40 (q2=q1+dq2)
41 dq3 = posj(25, 50, 40, 100, 30, 10) # q2에 대한 상대관절각 dq3 정의(q3=q2+dq3)
42
43 dq4 = posj(-20, -40, -20, -70, -40, 10) # q3에 대한 상대관절각 dq4 정의
44 (q4=q3+dq4)
45 dq5 = posj(-10, 10, 10, 40, -10, 30) # q4에 대한 상대관절각 dq5 정의
46 (q5=q4+dq5)
47
48
49 dqlist = [dq1, dq2, dq3, dq4, dq5]
50 # dq1~dq5를 상대경유점 집합으로 하는 리스트(dqlist ) 정의
51
52
53
54
55 movesj(dqlist, vel=30, acc=100, mod= DR_MV_MOD_REL )
56 # dqlist에 정의된 상대경유점 집합을 연결하는 스플라인 곡선을 최대속도
57 # 30(mm/sec), 최대가속도 100(mm/sec2)로 움직임 (CASE-1과 동일한 모션)
58
59

```

## 관련 명령어

- posj(J1=0, J2=0, J3=0, J4=0, J5=0, J6=0)(p. 29)
- set\_velj(vel)(p. 45)
- set\_accj(acc)(p. 47)
- amovesj()(p. 113)

### 3.3.6 movesx()

#### 기능

로봇이 현재 위치에서 pos\_list로 입력된 작업공간(task space)의 경유점들을 거쳐 목표위치(pos\_list의 마지막 경유점)까지 연결되는 스플라인 곡선경로를 따라 이동합니다.

입력된 속도/가속도는 경로 중 최대 속도/가속도이며 정속모션 옵션을 선택할 경우 조건에 따라 입력한 속도로 정속도의 모션을 수행합니다.

#### 인수

인수명	자료형	기본값	설명
pos_list	list (posx)	-	posx list
vel (v)	float	None	velocity 또는 velocity1, velocity2
	list (float[2])		
acc (a)	float	None	acceleration 또는 acceleration1, acceleration2
	list (float[2])		
time (t)	float	None	도달 시간 [sec]
ref	int	None	reference coordinate <ul style="list-style-type: none"> <li>• DR_BASE: base coordinate</li> <li>• DR_WORLD: world coordinate</li> <li>• DR_TOOL: tool coordinate</li> <li>• user coordinate: 사용자 정의</li> </ul>
mod	int	DR_MV_MOD_ABS	이동 기준 <ul style="list-style-type: none"> <li>• DR_MV_MOD_ABS: 절대</li> <li>• DR_MV_MOD_REL: 상대</li> </ul>
vel_opt	int	DR_MVS_VEL_NONE	속도 옵션 <ul style="list-style-type: none"> <li>• DR_MVS_VEL_NONE: 없음</li> <li>• DR_MVS_VEL_CONST: 등속</li> </ul>

#### ● 알아두기

- 단축 인수를 지원합니다. (v:vel, a:acc, t:time)
- vel이 None인 경우 \_global\_velx가 적용됩니다. (\_global\_velx 초기값은 0.0이며, set\_velx에 의해 설정 가능)
- acc이 None인 경우 \_global\_accx가 적용됩니다. (\_global\_accx 초기값은 0.0이며, set\_accx에 의해 설정 가능)
- vel에 하나의 인자를 입력한 경우(예를들어, vel=30) 입력된 인자는 모션의 선속도에 대응되며, 각속도는 선속도에 비례하여 결정됩니다.
- acc에 하나의 인자를 입력한 경우(예를들어, acc=60) 입력된 인자는 모션의 선가속도에 대응되며, 각각속도는 선가속도에 비례하여 결정됩니다.
- time을 지정할 경우 vel, acc를 무시하고 time 기준으로 처리됩니다.
- time이 None인 경우 0으로 처리됩니다.
- ref가 None인 경우 \_g\_coord가 적용됩니다. (\_g\_coord 초기값은 DR\_BASE이며, set\_ref\_coord 명령에 의해 설정 가능)
- mod가 DR\_MV\_MOD\_REL인 경우 pos\_list의 각 pos는 앞 선 pos에 대한 상대좌표로 정의됩니다. (pos\_list=[p1, p2, ..., p(n-1), p(n)]로 이루어질 때 p1은 시작점 대비 상대각도, p(n)은 p(n-1) 대비 상대좌표)
- 선행모션과 후행모션에 대한 온라인 블렌딩은 지원하지 않습니다.

#### 주의

vel\_opt=DR\_MVS\_VEL\_CONST 옵션(등속모션)을 선택할 경우 입력된 경유점 간 거리와 속도 조건에 따라 등속모션을 사용할 수 없을 수 있으며, 이 경우에 변속모션 (vel\_opt=DR\_MVS\_VEL\_NONE)으로 자동 전환됩니다.

#### 리턴

값	설명
0	성공
음수값	오류

#### 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시

예외	설명
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 #CASE 1) 절대좌표 입력 (mod= DR_MV_MOD_ABS)
2
3 P0 = posj(0,0,90,0,90,0)
4
5 movej(P0, v=30, a=30)
6
7 x0 = posx(600, 43, 500, 0, 180, 0) # posx 변수(공간좌표/자세) x0 정의
8
9 movel(x0, vel=100, acc=200) # 초기위치 x0로 line모션
10
11 x1 = posx(600, 600, 600, 0, 175, 0) # posx 변수(공간좌표/자세) x1 정의
12
13 x2 = posx(600, 750, 600, 0, 175, 0)
14
15 x3 = posx(150, 600, 450, 0, 175, 0)
16
17 x4 = posx(-300, 300, 300, 0, 175, 0)
18
19 x5 = posx(-200, 700, 500, 0, 175, 0)
20
21 x6 = posx(600, 600, 400, 0, 175, 0)
22
23
24
25 xlist = [x1, x2, x3, x4, x5, x6] # x1~x6를 경유점 집합으로 하는 리스트 xlist 정의
26
27
28
29 movesx(xlist, vel=[100, 30], acc=[200, 60], vel_opt=DR_MVS_VEL_NONE)
30
31 # 현재위치에서 시작하여 xlist에 정의된 경유점 집합을 연결하는 스플라인
32
33 # 곡선을 최대속도 100, 30(mm/sec, deg/sec), 최대가속도 200(mm/sec2),
34
35 # 60(deg/sec2)로 움직임
36
37 movesx(xlist, vel=[100, 30], acc=[200, 60], time=5, vel_opt=DR_MVS_VEL_CON
38 ST)

```

```

39 # 현재위치에서 시작하여 xlist에 정의된 경유점 집합을 연결하는 스플라인
40
41 # 곡선을 정속 100, 30(mm/sec, deg/sec)(가감속구간제외)로 움직임
42
43
44
45 #CASE 2) 상대좌표 입력 (mod= DR_MV_MOD_REL)
46
47 P0 = posj(0,0,90,0,90,0)
48
49 movej(P0)
50
51 x0 = posx(600, 43, 500, 0, 180, 0) # posx 변수(공간좌표/자세) x0 정의
52
53 movel(x0, vel=100, acc=200) # 초기위치 x0로 line모션
54
55 dx1 = posx(0, 557, 100, 0, -5, 0)
56
57 # x0에 대한 상대좌표 dx1 정의(x1=x0기준 dx1의 동차변환)
58
59 dx2 = posx(0, 150, 0, 0, 0, 0)
60
61 # x1에 대한 상대좌표 dx2 정의(x2=x1기준 dx2의 동차변환)
62
63 dx3 = posx(-450, -150, -150, 0, 0, 0)
64
65 # x2에 대한 상대좌표 dx3 정의(x3=x2기준 dx3의 동차변환)
66
67 dx4 = posx(-450, -300, -150, 0, 0, 0)
68
69 # x3에 대한 상대좌표 dx4 정의(x4=x3기준 dx4의 동차변환)
70
71 dx5 = posx(100, 400, 200, 0, 0, 0)
72
73 # x4에 대한 상대좌표 dx5 정의(x5=x4기준 dx5의 동차변환)
74
75 dx6 = posx(800, -100, -100, 0, 0, 0)
76
77 # x5에 대한 상대좌표 dx6 정의(x6=x5기준 dx6의 동차변환)
78
79
80
81 dxlist = [dx1, dx2, dx3, dx4, dx5, dx6]
82
83 # dx1~dx6를 경유점 집합으로 하는 리스트 dxlist 정의
84
85
86
87 movesx(dxlist, vel=[100, 30], acc=[200, 60], mod= DR_MV_MOD_REL, vel_opt=D
R_MVS_VEL_NONE)
88
89 # 현재위치에서 시작하여 dxlist에 정의된 상대 경유점 집합을 연결하는
90
91 # 스플라인 곡선을 최대속도 100(mm/sec), 30(deg/sec),

```

92  
93

# 최대가속도 200(mm/sec2), 60(deg/sec2)로 움직임 (CASE-1과 동일한 모션)

## 관련 명령어

- posx(x=0, y=0, z=0, A=0, B=0, C=0, ori\_type=None, sol=None, turn=None)(p. 30)
- set\_velx(vel1, vel2)(p. 48)
- (3.2.2-ko\_KR) set\_velx(vel)<sup>6</sup>
- set\_accx(acc1, acc2)(p. 50)
- (3.2.2-ko\_KR) set\_accx(acc)<sup>7</sup>
- set\_tcp(name)(p. 52)
- set\_ref\_coord(coord)(p. 53)
- amovesx()(p. 116)

### 3.3.7 moveb()

#### 기능

하나 이상의 경로 세그먼트(line 또는 circle)를 인자로 갖는 list를 받아 각 세그먼트를 설정된 radius로 블렌딩하여 등 속으로 이동합니다. 여기서 radius는 posb를 통해 설정 가능합니다.

#### 인수

인수명	자료형	기본값	설명
pos_list	list (posb)	-	posb list
vel (v)	float	None	velocity 또는 velocity1, velocity2
	list (float[2])		
acc (a)	float	None	acceleration 또는 acceleration1, acceleration2
	list (float[2])		
time (t)	float	None	도달 시간 [sec] • time 지정 시, vel, acc를 무시하고 time 기준으로 처리

<sup>6</sup> [https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/202473924/3.2.2-ko\\_KR+set\\_velx+vel](https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/202473924/3.2.2-ko_KR+set_velx+vel)

<sup>7</sup> [https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/202179059/3.2.2-ko\\_KR+set\\_accx+acc](https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/202179059/3.2.2-ko_KR+set_accx+acc)

인수명	자료형	기본값	설명
ref	int	None	<p>reference coordinate</p> <ul style="list-style-type: none"> <li>• DR_BASE: base coordinate</li> <li>• DR_WORLD: world coordinate</li> <li>• DR_TOOL: tool coordinate</li> <li>• user coordinate: 사용자 정의</li> </ul>
mod	int	DR_MV_MOD_ABS	<p>이동 기준</p> <ul style="list-style-type: none"> <li>• DR_MV_MOD_ABS: 절대</li> <li>• DR_MV_MOD_REL: 상대</li> </ul>
app_type	int	DR_MV_APP_NONE	<p>어플리케이션 연동 옵션</p> <ul style="list-style-type: none"> <li>• DR_MV_APP_NONE: 연동없음</li> <li>• DR_MV_APP_WELD: 용접연동</li> </ul>

### ❶ 알아두기

- 단축 인수를 지원합니다. (v:vel, a:acc, t:time)
- posb\_list는 최대 50개까지 입력할 수 있습니다.
- vel이 None인 경우 \_global\_velx가 적용됩니다. (\_global\_velx 초기값은 0.0이며, set\_velx에 의해 설정 가능)
- acc이 None인 경우 \_global\_accx가 적용됩니다. (\_global\_accx 초기값은 0.0이며, set\_accx에 의해 설정 가능)
- vel에 하나의 인자를 입력한 경우(예를들어, vel=30) 입력된 인자는 모션의 선속도에 대응되며, 각속도는 선속도에 비례하여 결정됩니다.
- acc에 하나의 인자를 입력한 경우(예를들어, acc=60) 입력된 인자는 모션의 선가속도에 대응되며, 각가속도는 선가속도에 비례하여 결정됩니다.
- time을 지정할 경우 vel, acc를 무시하고 time 기준으로 처리됩니다.
- time이 None인 경우 0으로 처리됩니다.
- ref가 None인 경우 \_g\_coord가 적용됩니다. (\_g\_coord 초기값은 DR\_BASE이며, set\_ref\_coord 명령에 의해 설정 가능)
- mod가 DR\_MV\_MOD\_REL인 경우 posb\_list의 각 pos는 앞선 pos에 대한 상대좌표로 정의됩니다.
- app\_type이 DR\_MV\_APP\_WELD인 경우에 이동속도는 vel인자의 입력값이 아닌 app\_weld\_set\_weld\_cond()에 입력된 속도설정으로 대체됩니다.

### ⚠ 주의

- posb에서 blending radius가 0인 경우, 사용자 입력 오류가 나타납니다.
- 연속된 Line-Line segment가 같은 방향을 가질 경우 Line의 중복입력으로 사용자 입력 오류가 나타납니다.
- 블렌딩 구간에서 조건에 따라 급격하게 방향전환이 발생하게 되는 경우 급가속을 방지하기 위해 사용자 입력오류가 나타납니다.

- 선행모션과 후행모션에 대한 온라인 블렌딩은 지원하지 않습니다.

## 리턴

값	설명
0	성공
음수값	오류

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 # Init Pose @ Jx1
2
3 Jx1 = posj(45,0,90,0,90,45)           #초기 Joint위치
4
5 X0 = posx(370, 420, 650, 0, 180, 0) #초기 Task위치

```

```

1 # CASE 1) ABSOLUTE
2
3 # Absolute Goal Poses
4
5 X1 = posx(370, 670, 650, 0, 180, 0)
6
7 X1a = posx(370, 670, 400, 0, 180, 0)
8
9 X1a2= posx(370, 545, 400, 0, 180, 0)

```

```

10 X1b = posx(370, 595, 400, 0, 180, 0)
11 X1b2= posx(370, 670, 400, 0, 180, 0)
12 X1c = posx(370, 420, 150, 0, 180, 0)
13 X1c2= posx(370, 545, 150, 0, 180, 0)
14 X1d = posx(370, 670, 275, 0, 180, 0)
15 X1d2= posx(370, 795, 150, 0, 180, 0)
16
17
18
19
20
21
22
23
24
25 seg11 = posb(DR_LINE, X1, radius=20)
26 seg12 = posb(DR_CIRCLE, X1a, X1a2, radius=20)
27 seg14 = posb(DR_LINE, X1b2, radius=20)
28 seg15 = posb(DR_CIRCLE, X1c, X1c2, radius=20)
29 seg16 = posb(DR_CIRCLE, X1d, X1d2, radius=20)
30
31 b_list1 = [seg11, seg12, seg14, seg15, seg16]
32
33 # 마지막경유점(seg16)의 blending radius는 무시됨
34
35
36
37 movej(Jx1, vel=30, acc=60, mod=DR_MV_MOD_ABS)
38
39
40
41 # 초기각도(Jx1)로 Joint모션
42
43 movel(X0, vel=150, acc=250, ref=DR_BASE, mod=DR_MV_MOD_ABS)
44
45 #초기위치(X0)로 line모션
46
47 moveb(b_list1, vel=150, acc=250, ref=DR_BASE, mod=DR_MV_MOD_ABS)
48
49 # 현재위치에서 시작하여 seg11(LINE), seg12(CIRCLE), seg14(LINE),
50 # seg15(CIRCLE), seg16(CIRCLE)으로 이루어진 궤적을 속도 150(mm/sec)를
51 # 유지하며(가감속구간 제외) 움직임 (최종point는 X1d2). 각 segment의 끝점
52 # (X1, X1a2, X1b2, X1c2, X1d2)에서 20mm 거리에 도달하면 다음 segment로
53 # blending이 시작됨
54
55
56
57
58
59

```

1

# CASE 2) RELATIVE

```

2
3 # Relative Goal Poses
4
5 dX1 = posx(0, 250, 0, 0, 0, 0)
6
7 dX1a = posx(0, 0, -150, 0, 0, 0)
8
9 dX1a2= posx(0, -125, 0, 0, 0, 0)
10
11 dX1b = posx(0, 50, 0, 0, 0, 0)
12
13 dX1b2= posx(0, 75, 0, 0, 0, 0)
14
15 dX1c = posx(0, -250, -250, 0, 0, 0)
16
17 dX1c2= posx(0, 125, 0, 0, 0, 0)
18
19 dX1d = posx(0, 125, 125, 0, 0, 0)
20
21 dX1d2= posx(0, 125, -125, 0, 0, 0)
22
23
24
25 dseg11 = posb(DR_LINE, dX1, radius=20)
26
27 dseg12 = posb(DR_CIRCLE, dX1a, dX1a2, radius=20)
28
29 dseg14 = posb(DR_LINE, dX1b2, radius=20)
30
31 dseg15 = posb(DR_CIRCLE, dX1c, dX1c2, radius=20)
32
33 dseg16 = posb(DR_CIRCLE, dX1d, dX1d2, radius=20)
34
35 db_list1 = [dseg11, dseg12, dseg14, dseg15, dseg16]
36
37 # 마지막경유점(dseg16)의 blending radius는 무시됨
38
39
40
41 movej(Jx1, vel=30, acc=60, mod=DR_MV_MOD_ABS)
42
43 # 초기각도(Jx1)로 Joint모션
44
45 movel(X0, vel=150, acc=250, ref=DR_BASE, mod=DR_MV_MOD_ABS)
46
47 #초기위치(X0)로 line모션
48
49 moveb(b_list1, vel=150, acc=250, ref=DR_BASE, mod=DR_MV_MOD_ABS)
50
51     # 현재위치에서 시작하여 상대위치로 정의된 dseg11(LINE), dseg12(CIRCLE),
52
53     # dseg14(LINE), dseg15(CIRCLE), dseg16(CIRCLE)으로 이루어진 궤적을
54
55     # 속도 150(mm/sec)를 유지하며(가감속구간제외) 움직임 (최종point는 X1d2).
```

```

56
57      # 각 segment의 끝점(x1, x1a2, x1b2, x1c2, x1d2)에서 20mm 거리에
58
59      # 도달하면 다음 segment로 blending이 시작됨 (경로는 CASE#1과 동일)

```

## 관련 명령어

- posb(seg\_type, posx1, posx2=None, radius=0)(p. 35)
- set\_velx(vel1, vel2)(p. 48)
- (3.2.2-ko\_KR) set\_velx(vel)<sup>8</sup>
- set\_accx(acc1, acc2)(p. 50)
- (3.2.2-ko\_KR) set\_accx(acc)<sup>9</sup>
- set\_tcp(name)(p. 52)
- set\_ref\_coord(coord)(p. 53)
- amoveb()(p. 119)

### 3.3.8 move\_spiral()

#### 기능

지정한 좌표계(ref) 상에서 현재위치를 중심으로 지정축(axis)에 수직한 평면 위의 나선궤적을 따르는 모션을 수행합니다. 이동거리(lmax)를 추가로 입력하면 원뿔의 꼭지점으로부터 시작하여 주위를 따라 도는 궤적의 모션을 수행합니다.

#### 인수

인수명	자료형	기본값	범위	설명
rev	float	10	rev > 0	총 회전수 [revolution]
rmax	float	10	rmax > 0	spiral 최종 반경 [mm]
lmax	float	0		axis 방향으로 이동하는 거리 [mm]
vel (v)	float	None		velocity
acc (a)	float	None		acceleration
time (t)	float	None	time ≥ 0	총 수행시간 <sec>

<sup>8</sup> [https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/202473924/3.2.2-ko\\_KR+set\\_velx+vel](https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/202473924/3.2.2-ko_KR+set_velx+vel)

<sup>9</sup> [https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/202179059/3.2.2-ko\\_KR+set\\_accx+acc](https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/202179059/3.2.2-ko_KR+set_accx+acc)

인수명	자료형	기본값	범위	설명
axis	int	DR_AXIS_Z	-	axis <ul style="list-style-type: none"> <li>• DR_AXIS_X: x축</li> <li>• DR_AXIS_Y: y축</li> <li>• DR_AXIS_Z: z축</li> </ul>
ref	Int	DR_TOOL	-	reference coordinate <ul style="list-style-type: none"> <li>• DR_BASE : base coordinate</li> <li>• DR_WORLD : world coordinate</li> <li>• DR_TOOL : tool coordinate</li> <li>• user coordinate : 사용자 정의</li> </ul>

### ❶ 알아두기

- 단축 인수를 지원합니다. (v:vel, a:acc, t:time)
- rev는 spiral 모션의 총 회전수를 의미합니다.
- rmax는 spiral 모션의 최대 반경을 의미합니다.
- lmax는 모션 동안 axis 방향으로 병진하는 거리를 의미합니다. 단, 음수인 경우 -axis 방향 병진합니다.
- vel은 spiral 모션의 이동 속도를 의미합니다.
- vel이 None인 경우, \_global\_velx의 첫째 값(병진 속도)이 적용됩니다. (\_global\_velx 초기값은 0.0이며, set\_velx에 의해 설정 가능)
- acc는 spiral 모션의 이동 가속도를 의미합니다.
- acc가 None인 경우, \_global\_accx 첫째 값(병진 가속도)이 적용됩니다. (\_global\_accx 초기값은 0.0이며, set\_accx에 의해 설정 가능)
- time을 지정할 경우 vel, acc를 무시하고 time 기준으로 처리됩니다.
- time이 None인 경우 0으로 처리됩니다.
- axis는 Spiral 모션이 정의하는 평면에 수직인 축을 정의합니다.
- ref는 spiral 모션이 정의하는 기준 좌표계를 의미합니다.
- 선행모션과 후행모션에 대한 온라인 블렌딩은 지원하지 않습니다.

### ⚠ 주의

- 경로 생성 시 Spiral 경로에 의한 회전각 가속도를 연산하여 값이 큰 경우 안정적인 모션을 위하여 에러가 발생나타날 수 있습니다.  
이 경우 vel, acc 또는 time 값을 작게 조정하는 것을 권장합니다.

## 리턴

값	설명
0	성공
음수값	오류

## 예외

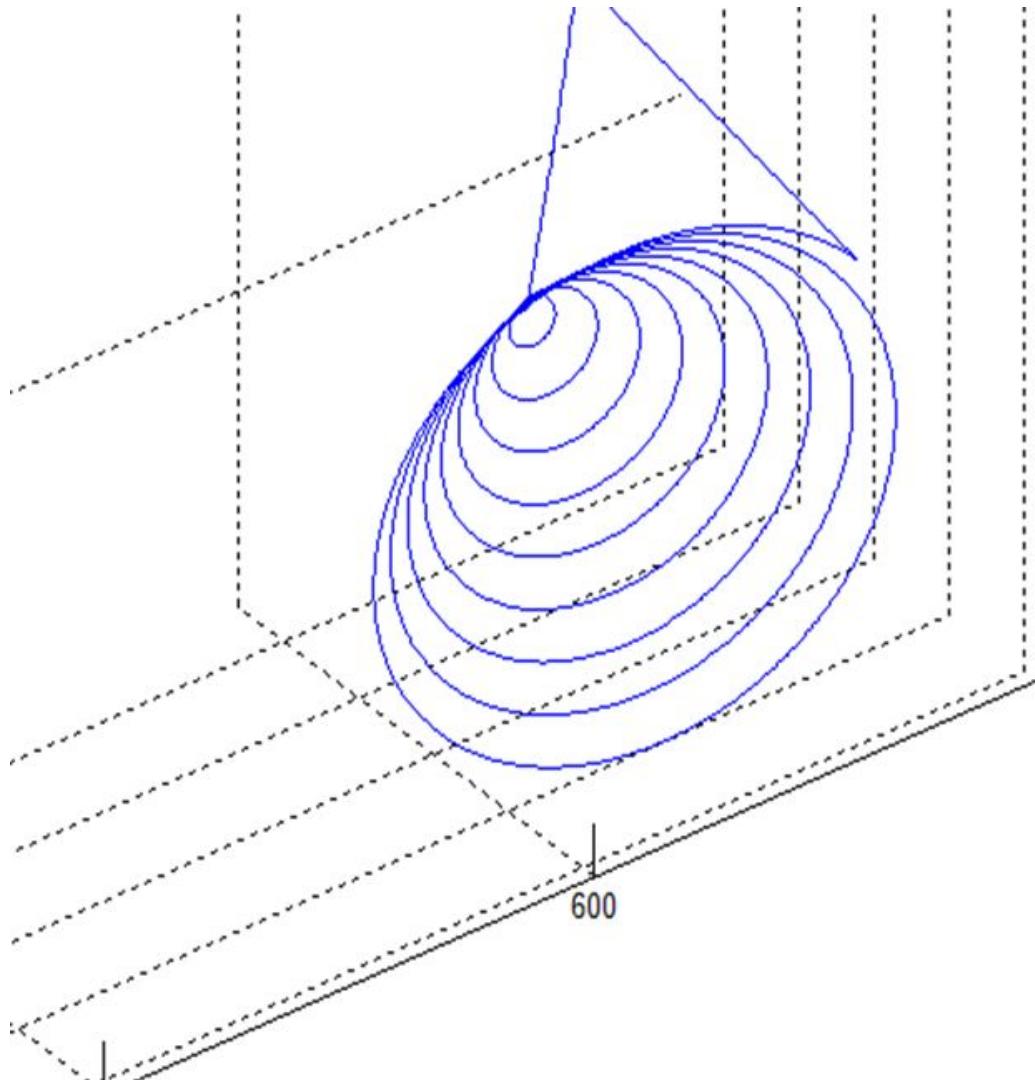
예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 # hole search
2
3 #(초기 위치로부터 Tool-Z 방향의 회전중심으로, Tool-X/Y 평면의 0에서 20mm 반경(rmax)으로
4   9.5회전(rev) 함과 동시에 Tool-Z 방향으로 50mm(lmax) 이동하는 spiral 궤적을 20초에 완
5   료하는 모션)
6
7 J00 = posj(0,0,90,0,60,0)
8
9 movej(J00,vel=30,acc=30) # 초기 자세로 Joint 이동
10
11 move_spiral(rev=9.5,rmax=20.0,lmax=50.0,time=20.0,axis=DR_AXIS_Z,ref=DR_TO
    OL)

```



## 관련 명령어

- `set_velx(vel1, vel2)`(p. 48)
- `(3.2.2-ko_KR) set_velx(vel)`<sup>10</sup>
- `set_accx(acc1, acc2)`(p. 50)
- `(3.2.2-ko_KR) set_accx(acc)`<sup>11</sup>
- `set_tcp(name)`(p. 52)
- `set_ref_coord(coord)`(p. 53)
- `amove_spiral()`(p. 124)

<sup>10</sup> [https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/202473924/3.2.2-ko\\_KR+set\\_velx+vel](https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/202473924/3.2.2-ko_KR+set_velx+vel)

<sup>11</sup> [https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/202179059/3.2.2-ko\\_KR+set\\_accx+acc](https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/202179059/3.2.2-ko_KR+set_accx+acc)

### 3.3.9 move\_periodic()

#### 기능

현재 위치에서 시작하는 상대 모션으로 입력된 기준 좌표계(ref)의 각 축(병진 및 회전)에 대한 Sine 함수 기반으로 주기 모션을 수행합니다. 각 axis 별 모션의 특성은 amp(amplitude)와 period에 의해 결정되고, 가감속 시간과 총 모션 시간은 주기, 반복, 횟수에 의해 설정됩니다.

#### 인수

인수명	자료형	기본값	범위	설명
amp	list (float[6])	-	0≤amp	Amplitude(-amp에서 +amp사이 모션) [mm] or [deg]
period	float or list (float[6])		0≤period	period(1주기 소요 시간)[sec]
atime	float	0.0	0≤atime	Acc-, dec- time [sec]
repeat	int	1	> 0	반복 횟수
ref	int	DR_TOOL	-	reference coordinate <ul style="list-style-type: none"> <li>• DR_BASE : base coordinate</li> <li>• DR_WORLD : world coordinate</li> <li>• DR_TOOL : tool coordinate</li> <li>• user coordinate : 사용자 정의</li> </ul>

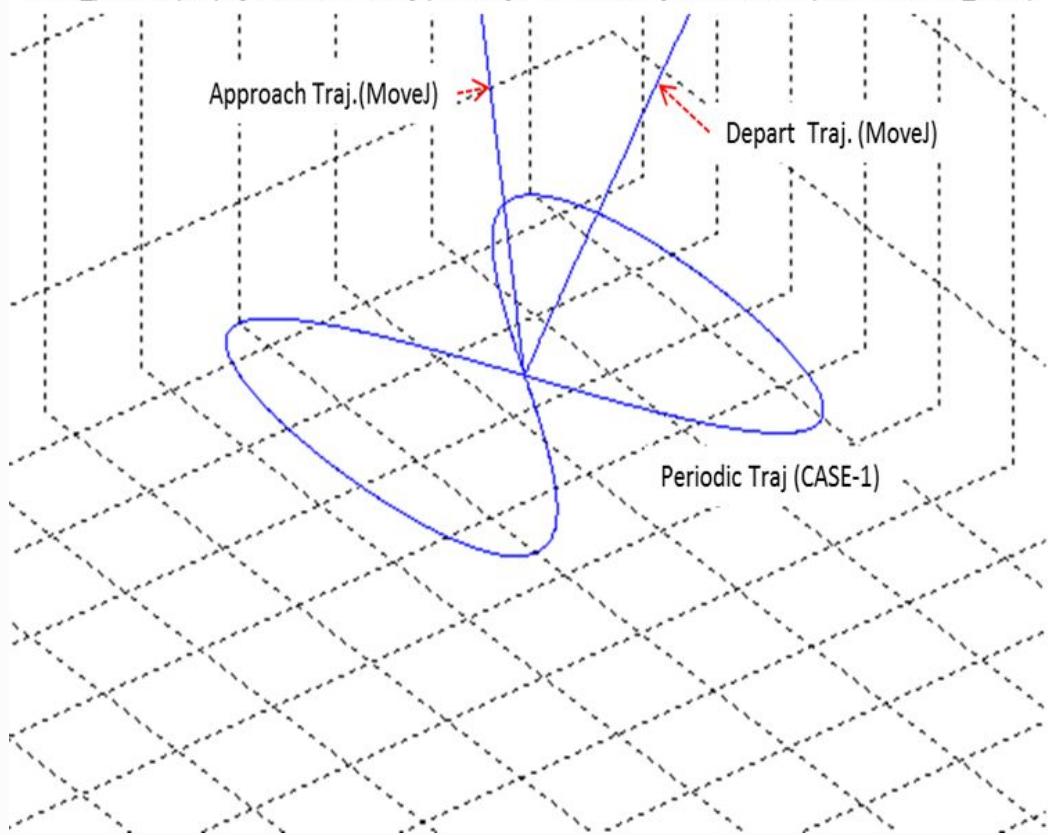
#### 알아두기

- amp는 진폭(amplitude)을 의미하며, 각 축(x, y, z, rx, ry, rz) 별로 amp를 값으로 하는 6개 원소의 list 형태로 입력해야 합니다. 단, 주기 모션을 진행하지 않는 축 방향은 amp를 0으로 입력해야 합니다.
- period는 해당 방향 모션의 1회 반복 시간을 의미하며, 각 축(x, y, z, rx, ry, rz) 별 period를 값으로 하는 총 6개 원소의 list 형태로 입력하거나 대표값을 입력해야합니다.
- atime은 주기모션의 시작과 끝의 가속 및 감속 시간을 의미합니다. 입력된 가감속시간과 최대주기 \*1/4 중 큰 값이 적용됩니다. 입력된 가감속 시간이 전체모션시간의 1/2을 초과하는 경우 에러가 발생합니다.

- repeat은 가장 큰 period 값을 가지는 축(기준 축)의 반복 횟수를 정의하며, 이에 따라 총 모션 시간이 결정됩니다. 나머지 축의 반복 횟수는 모션 시간에 따라 자동 결정됩니다.
- 모션이 정상 종료되는 경우 종료 위치가 시작 위치와 일치하게 하도록 나머지 축 모션은 기준 축 모션이 종료되기 전에 먼저 종료될 수 있습니다. 모든 축의 모션이 동시에 종료되지 않는 경우 감속구간에서의 경로는 이전 경로에서 벗어나게 됩니다. 관련한 사항은 아래 이미지를 참조하십시오

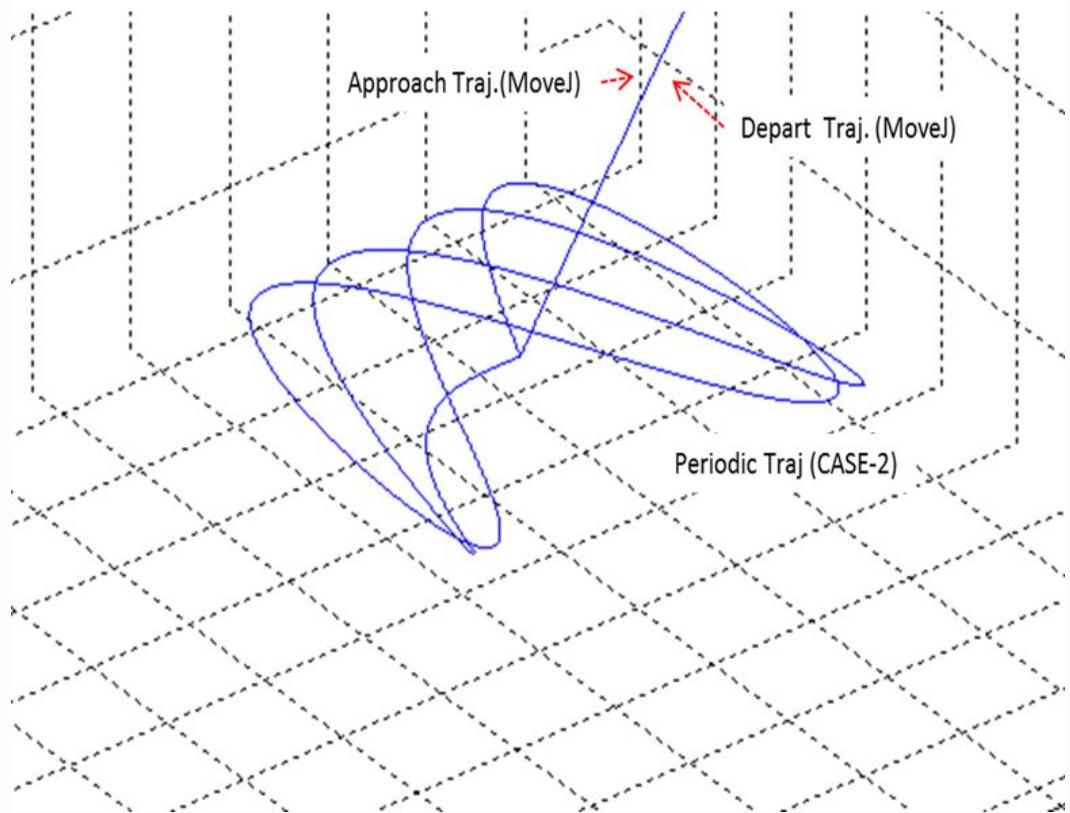
**CASE-1) All-axis motions end at the same time**

```
move_periodic(amp=[100,100,0,0,0,0], period=[3.2,1.6,0,0,0,0], atime=3.1, repeat=2, ref=DR_BASE)
```



**CASE-2) Diff-axis motions end individually**

```
move_periodic(amp=[100,100,0,0,0,0], period=[3.2,1.5,0,0,0,0], atime=0, repeat=2, ref=DR_BASE)
```



- ref는 반복 모션의 기준 좌표계를 의미합니다.
- 모션명령 수행 시 최대속도 에러가 발생하는 경우 다음의 식을 참조하여 진폭 및 주기를 조정할 것을 제안합니다.  
**최대속도=진폭(amp)\*2\*pi(3.14)/주기(period)**  
(예, 진폭=10mm, 주기=1초인 경우 최대속도=62.83mm/sec)
- 선행모션과 후행모션에 대한 온라인 블렌딩은 지원하지 않습니다.

**리턴**

값	설명
0	성공
음수값	오류

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 P0 = posj(0,0,90,0,90,0)
2 movej(P0)
3
4 #1
5 move_periodic(amp =[10,0,0,0,30,0], period=1.0, atime=0.2, repeat=5, ref=D
R_TOOL)
6 # Tool 좌표계 x축(10mm 진폭, 1초 주기) 모션과 y회전축(진폭 30deg, 1초 주기)
7 # 모션이 총 5회 반복 수행
8
9 #2
10 move_periodic(amp =[10,0,20,0,0.5,0], period=[1,0,1.5,0,0,0], atime=0.5,
11 repeat=3, ref=DR_BASE)
12 # BASE 좌표계 x축(10mm 진폭, 1초 주기), z축(20mm 진폭, 1.5초 주기) 모션이
13 # 총 3회 반복 수행됨, y회전축 모션은 period가 ‘zero(0)’이므로 미수행
14 # z축 모션의 주기가 크므로 총 모션 시간은 약 5.5초(1.5초*3회 + 가감속 1초)
# 이며, x축은 4.5회 반복 수행

```

## 관련 명령어

- [set\\_ref\\_coord\(coord\)\(p. 53\)](#)
- [amove\\_periodic\(\)\(p. 128\)](#)

### 3.3.10 move\_home()

#### 기능

기계적 홈 또는 사용자정의 홈 위치로 조인트모션으로 이동하여 호밍을 수행합니다. 입력인자[target]에 따라 시스템에 정의된 기계적 홈 또는 사용자설정 홈으로 이동합니다.

**⚠ 주의**

Homming 완료 후 Move 명령어 사용 시, Homming 이후에 wait 명령어 사용하기를 권장함

#### 인수

인수명	자료형	기본값	범위	설명
target	int	-		목표하는 홈자세 <ul style="list-style-type: none"> <li>• DR_HOME_TARGET_MECHANIC : 기계적 홈, 조인트각도 (0,0,0,0,0,0)</li> <li>• DR_HOME_TARGET_USER : 사용자설정 홈위치</li> </ul>

**ℹ 알아두기**

- 호밍동작은 크게 두 동작으로 나뉘어 순차적으로 수행됩니다.
  - a. 호밍위치로 시스템에 지정된속도로 이동
  - b. 정밀하게 홈위치를 찾는 동작
- 호밍동작 중 주변에 충돌위험이 있지 않도록 안전을 확보하여야 합니다.

#### 리턴

값	설명
0	성공
음수값	오류

#### 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시

예외	설명
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 move_home(DR_HOME_TARGET_USER) # 사용자가 설정한 홈 위치로 이동
2
3 P0 = posj(0,0,90,0,90,0)
4 movej(P0)

```

## 3.4 비동기 모션

### 3.4.1 amovej()

#### 기능

비동기(async.)방식의 movej로 블렌딩을 위한 radius 인자를 갖지 않는 점을 제외하고 movej와 동일하게 작동합니다. 그러나 해당 명령어는 async 방식의 모션 명령어로 모션 시작과 동시에 다음 명령어를 수행합니다.

- posj가 입력될 경우 기존처럼 amovej로 동작
- posx가 입력될 경우 기존 amovejx처럼 동작 (posx 입력할 경우 ref 변수 추가 입력 가능)

#### 비교

- movej(pos): 현재 위치에서 출발하여 pos에 도달(정지)한 후에 다음 명령 수행
- amovej(pos): 현재 위치에서 출발하여 pos 도달(정지) 여부와 관계없이 즉시 다음 명령 수행

## 인수

인수명	자료형	기본값	설명
pos	posj	-	posj 또는 joint angle list posx
	list (float[6])		
	posx		
vel (v)	float	None	velocity(모든 축에 동일) 또는 velocity(축별 velocity)
	list (float[6])		
acc (a)	float	None	acceleration(모든 축에 동일) 또는 acceleration(축별 acceleration)
	list (float[6])		
time (t)	float	None	도달 시간 [sec]
mod	int	DR_MV_MOD_ABS	이동 기준 <ul style="list-style-type: none"> <li>DR_MV_MOD_ABS: 절대</li> <li>DR_MV_MOD_REL: 상대</li> </ul>
ra	int	DR_MV_RA_DUPLICATE	Reactive motion mode <ul style="list-style-type: none"> <li>DR_MV_RA_DUPLICATE: duplicate</li> <li>DR_MV_RA_OVERRIDE: override</li> </ul>
ref	int	None	reference coordinate <ul style="list-style-type: none"> <li>DR_BASE: base coordinate</li> <li>DR_WORLD: world coordinate</li> <li>DR_TOOL: tool coordinate</li> <li>user coordinate: 사용자 정의</li> </ul>

### 알아두기

- 단축 인수를 지원합니다. (v:vel, a:acc, t:time)
- vel이 None인 경우, \_global\_velj가 적용됩니다. (\_global\_velj 초기값은 0.0이며, set\_velj에 의해 설정 가능)
- acc이 None인 경우, \_global\_accj가 적용됩니다. (\_global\_accj 초기값은 0.0이며, set\_accj에 의해 설정 가능)
- time을 지정할 경우 vel, acc를 무시하고 time 기준으로 처리됩니다.

- time이 None인 경우 0으로 처리됩니다.
- 옵션 ra 및 vel/acc에 따른 blending 시의 경로는 movej() 모션 설명을 참조할 것

## 리턴

값	설명
0	성공
음수값	오류

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 #예제 1. q0로 모션 시작 3초 후에 q1으로 움직여 모션 정지 후 q99으로 이동
2
3 q0 = posj(0, 0, 90, 0, 90, 0)
4
5 amovej (q0, vel=10, acc=20) # q0로 모션 및 즉시 다음 명령 수행
6
7 wait(3) # 3초간 프로그램 일시 중지(모션은 진행 중)
8
9 q1 = posj(0, 0, 0, 0, 90, 0)
10
11 amovej (q1, vel=10, acc=20)
12
13 # q0 모션을 유지(ra 인자 생략 시 DUPLICATE blending)하며 q1으로 중첩
14
15 # blending하는 모션 및 즉시 다음 명령 수행

```

```

16
17     mwait(0)    # 모션이 종료할 때까지 프로그램 일시 중지
18
19     q99 = posj(0, 0, 0, 0, 0, 0)
20
21     movej (q99, vel=10, acc=20) # q99으로 조인트 모션

```

## 관련 명령어

- posj(J1=0, J2=0, J3=0, J4=0, J5=0, J6=0)(p. 29)
- set\_velj(vel)(p. 45)
- set\_accj(acc)(p. 47)
- mwai~~t(time=0)~~(p. 132)
- movej()(p. 57)

## 3.4.2 amovel()

### 기능

비동기(async.)방식의 move()모션으로 블렌딩을 위한 radius인자를 갖지 않는 점을 제외하고 move()와 동일하게 작동합니다. 그러나 해당 명령어는 async 방식의 모션명령어로 모션 종료를 기다리지 않고 다음 명령어를 수행합니다.

#### 비교

- move(pos) : 현재위치에서 출발하여 pos에 도달(정지)한 후에 다음 명령 수행
- amovel(pos) : 현재위치에서 출발하여 pos 도달(정지)여부와 관계없이 즉시 다음 명령 수행

### 인수

인수명	자료형	기본값	설명
pos	posx	-	posx 또는 position list
	list (float[6])		
vel (v)	float	None	velocity 또는 velocity1, velocity2
	list (float[2])		
acc (a)	float	None	acceleration 또는 acceleration1, acceleration2
	list (float[2])		

인수명	자료형	기본값	설명
time (t)	float	None	도달 시간 [sec] • time 지정 시, vel, acc를 무시하고 time 기준으로 처리
ref	int	None	reference coordinate • DR_BASE : base coordinate • DR_WORLD : world coordinate • DR_TOOL : tool coordinate • user coordinate: 사용자 정의
mod	int	DR_MV_MOD_ABS	이동 기준 • DR_MV_MOD_ABS: 절대 • DR_MV_MOD_REL: 상대
ra	int	DR_MV_RA_DUPLICATE	Reactive motion mode • DR_MV_RA_DUPLICATE: duplicate • DR_MV_RA_OVERRIDE: override
app_type	int	DR_MV_APP_NONE	어플리케이션 연동 옵션 • DR_MV_APP_NONE: 연동없음 • DR_MV_APP_WELD: 용접연동

### ❶ 알아두기

- 단축 인수 지원(v:vel, a:acc, t:time)
- vel가 None인 경우, \_global\_velx 적용(\_global\_velx 초기값은 0.0이며, set\_velx에 의해 설정 가능)
- acc가 None인 경우, \_global\_accx 적용(\_global\_accx 초기값은 0.0이며, set\_accx에 의해 설정 가능)
- vel에 하나의 인자를 입력한 경우(예를들어, vel=30) 입력된 인자는 모션의 선속도에 대응되며, 각속도는 선속도에 비례하여 결정됩니다.
- acc에 하나의 인자를 입력한 경우(예를들어, acc=60) 입력된 인자는 모션의 선가속도에 대응되며, 각가속도는 선가속도에 비례하여 결정됩니다.
- time을 지정할 경우 vel, acc를 무시하고 time 기준으로 처리됩니다.
- time이 None인 경우 0으로 처리됩니다.
- ref가 None인 경우, \_g\_coord 적용(\_g\_coord 초기값은 DR\_BASE이며, set\_ref\_coord 명령에 의해 설정 가능)
- 옵션 ra 및 vel/acc에 따른 blending 시의 경로는 movel() 모션 설명을 참조할 것
- app\_type이 DR\_MV\_APP\_WELD인 경우에 이동속도는 vel인자의 입력값이 아닌 app\_weld\_set\_weld\_cond()에 입력된 속도설정으로 대체됩니다.

## 리턴

값	설명
0	성공
음수값	오류

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 #예제 1. x1으로 모션시작 후 2초 후에 D-Out
2
3 j0 = posj(-148,-33,-54,180,92,32)
4
5 movej(j0, v=30, a=30)
6
7 x1 = posx(784, 543, 570, 0, 180, 0)
8
9 amovel (x1, vel=100, acc=200) # x1으로 모션 및 즉시 다음명령 수행
10
11 wait(2) # 2초간 프로그램 일시중지 (모션은 진행 중)
12
13 set_digital_output(1 , 1) # D-Out(1번채널) ON
14
15 mwait(0) # 모션이 종료할 때까지 프로그램 일시중지

```

## 관련 명령어

- posx(x=0, y=0, z=0, A=0, B=0, C=0, ori\_type=None, sol=None, turn=None)(p. 30)
- set\_velx(vel1, vel2)(p. 48)
- (3.2.2-ko\_KR) set\_velx(vel)<sup>12</sup>
- set\_accx(acc1, acc2)(p. 50)
- (3.2.2-ko\_KR) set\_accx(acc)<sup>13</sup>
- set\_tcp(name)(p. 52)
- set\_ref\_coord(coord)(p. 53)
- mwait(time=0)(p. 132)
- move1()(p. 62)

### 3.4.3 amovejx()

#### 기능

비동기(async.)방식의 movejx모션으로 블렌딩을 위한 radius인자를 갖지 않는 점을 제외하고 movejx와 동일하게 작동합니다. 그러나 해당 명령어는 async 방식의 모션명령어로 모션 종료를 기다리지 않고 다음 명령어를 수행합니다.

#### 비교

- movejx(pos): 현재 위치에서 출발하여 pos에 도달(정지)한 후에 다음 명령 수행
- amovejx(pos): 현재 위치에서 출발하여 pos 도달(정지) 여부와 관계없이 즉시 다음 명령 수행

#### 인수

인수명	자료형	기본값	설명
pos	posx	-	posx 또는 position list
	list (float[6])		
vel (v)	float	None	velocity(모든 축에 동일) 또는 velocity(축별 velocity)
	list (float[6])	None	
acc (a)	float	None	acceleration(모든 축에 동일) 또는 acceleration(축별 acceleration)
	list (float[6])	None	

<sup>12</sup> [https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/202473924/3.2.2-ko\\_KR+set\\_velx+vel](https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/202473924/3.2.2-ko_KR+set_velx+vel)

<sup>13</sup> [https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/202179059/3.2.2-ko\\_KR+set\\_accx+acc](https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/202179059/3.2.2-ko_KR+set_accx+acc)

인수명	자료형	기본값	설명
time (t)	float	None	도달 시간 [sec]
ref	int	None	reference coordinate <ul style="list-style-type: none"> <li>• DR_BASE: base coordinate</li> <li>• DR_WORLD : world coordinate</li> <li>• DR_TOOL: tool coordinate</li> <li>• user coordinate: 사용자 정의</li> </ul>
mod	int	DR_MV_MOD_ABS	이동 기준 <ul style="list-style-type: none"> <li>• DR_MV_MOD_ABS: 절대</li> <li>• DR_MV_MOD_REL: 상대</li> </ul>
ra	int	DR_MV_RA_DUPLICATE	Reactive motion mode <ul style="list-style-type: none"> <li>• DR_MV_RA_DUPLICATE: duplicate</li> <li>• DR_MV_RA_OVERRIDE: override</li> </ul>
sol	int	0	Solution space

### ● 알아두기

- 단축 인수를 지원합니다. (v:vel, a:acc, t:time)
- vel이 None인 경우 \_global\_velj가 적용됩니다. (\_global\_velj 초기값은 0.0이며, set\_velj에 의해 설정 가능합니다.)
- acc가 None인 경우 \_global\_accj가 적용됩니다. (\_global\_accj 초기값은 0.0이며, set\_accj에 의해 설정 가능합니다.)
- time을 지정할 경우 vel, acc를 무시하고 time 기준으로 처리됩니다.
- time이 None인 경우 0으로 처리됩니다.
- time을 지정할 경우 vel, acc를 무시하고 time 기준으로 처리됩니다.
- time이 None인 경우 0으로 처리됩니다.
- ref가 None인 경우 \_g\_coord가 적용됩니다. \_g\_coord의 초기값은 DR\_BASE이며, set\_ref\_coord 명령에 의해 설정이 가능합니다.
- 옵션 ra 와 vel/acc에 따른 블렌딩 상태의 경로는 movej() 모션 설명을 참조하십시오.

### ⚠ 주의

상대모션으로 입력하는 경우(mod=DR\_MV\_MOD\_REL), 진행중인 모션에 블렌딩 할 수 없으며 movej() 또는 movel()을 이용하여 블렌딩하는 것을 권장합니다.

## 리턴

값	설명
0	성공
음수값	오류

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 #예제 1. x1으로 조인트모션 시작 후 2초 후에 D-Out
2
3 p0 = posj(-148,-33,-54,180,92,32)
4
5 movej(p0, v=30, a=30)
6
7 x1 = posx(784, 443, 770, 0, 180, 0)
8
9 amovejx (x1, vel=100, acc=200, sol=2)    # x1으로 조인트모션 및 즉시 다음명령 수행
10
11 wait(2)          # 2초간 프로그램 일시중지 (모션은 진행 중)
12
13 set_digital_output(1 , 1)      # D-Out(1번채널) ON
14
15 mwait(0)

```

## 관련 명령어

- posx(x=0, y=0, z=0, A=0, B=0, C=0, ori\_type=None, sol=None, turn=None)(p. 30)
- set\_velj(vel)(p. 45)
- set\_accj(acc)(p. 47)
- get\_current\_posx(ref, ori\_type)(p. 168)
- mwait(time=0)(p. 132)
- movejx()(p. 67)

### 3.4.4 amovec()

#### 기능

비동기(async.)방식의 movec모션으로 블렌딩을 위한 radius인자를 갖지 않는 점을 제외하고 movec와 동일하게 작동합니다. 그러나 해당 명령어는 async 방식의 모션명령어로서 모션 종료를 기다리지 않고 다음 명령어를 수행합니다.

#### 비교

- movec(pos1, pos2): 현재위치에서 출발하여 pos2에 도달(정지)한 후에 다음 명령 수행
- amovec(pos1, pos2): 현재위치에서 출발하여 pos2 도달(정지)여부와 관계없이 즉시 다음 명령 수행

#### 인수

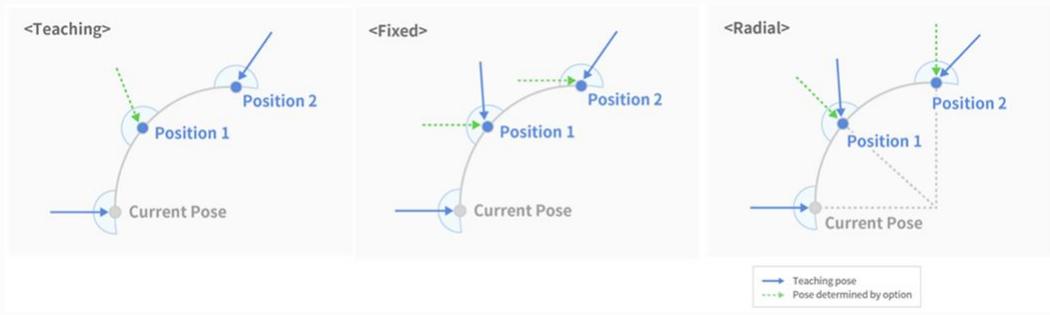
인수명	자료형	기본값	설명
pos	posx	-	posx 또는 position list
	list (float[6])		
pos2	posx	-	posx 또는 position list
	list (float[6])		
vel (v)	float	None	velocity 또는 velocity1, velocity2
	list (float[2])		
acc (a)	float	None	acceleration 또는 acceleration1, acceleration2
	list (float[2])		

인수명	자료형	기본값	설명
time (t)	float	None	도달 시간 [sec]
ref	int	None	reference coordinate <ul style="list-style-type: none"> <li>• DR_BASE: base coordinate</li> <li>• DR_WORLD: world coordinate</li> <li>• DR_TOOL: tool coordinate</li> <li>• user coordinate: 사용자 정의</li> </ul>
mod	int	DR_MV_MOD_ABS	이동 기준 <ul style="list-style-type: none"> <li>• DR_MV_MOD_ABS: 절대</li> <li>• DR_MV_MOD_REL: 상대</li> </ul>
angle (an)	float list (float[2])	None	angle 또는 angle1, angle2
ra	int	DR_MV_RA_DUPLICATE	Reactive motion mode <ul style="list-style-type: none"> <li>• DR_MV_RA_DUPLICATE: duplicate</li> <li>• DR_MV_RA_OVERRIDE: override</li> </ul>
ori	int	DR_MV_ORI_TEACH	이동자세 선택 <ul style="list-style-type: none"> <li>• DR_MV_ORI_TEACH: 교시자세</li> <li>• DR_MV_ORI_FIXED: 고정자세</li> <li>• DR_MV_ORI_RADIAL: 원주구속자세</li> </ul>
app_type	int	DR_MV_APP_NONE	어플리케이션 연동 옵션 <ul style="list-style-type: none"> <li>• DR_MV_APP_NONE: 연동없음</li> <li>• DR_MV_APP_WELD: 용접연동</li> </ul>

### ❶ 알아두기

- 단축 인수를 지원합니다. (v:vel, a:acc, t:time, angle:an)
- vel이 None인 경우 \_global\_velx가 적용됩니다. (\_global\_velx 초기값은 0.0이며, set\_velx에 의해 설정 가능)
- acc가 None인 경우 \_global\_accx가 적용됩니다. (\_global\_accx 초기값은 0.0이며, set\_accx에 의해 설정 가능)
- vel에 하나의 인자를 입력한 경우(예를들어, vel=30) 입력된 인자는 모션의 선속도에 대응되며, 각속도는 선속도에 비례하여 결정됩니다.
- acc에 하나의 인자를 입력한 경우(예를들어, acc=60) 입력된 인자는 모션의 선가속도에 대응되며, 각가속도는 선가속도에 비례하여 결정됩니다.

- time을 지정할 경우 vel, acc를 무시하고 time 기준으로 처리됩니다.
  - time이 None인 경우 0으로 처리됩니다.
  - ref가 None인 경우, \_g\_coord가 적용됩니다. (\_g\_coord 초기값은 DR\_BASE이며, set\_ref\_coord 명령에 의해 설정 가능)
  - mod가 DR\_MV\_MOD\_REL인 경우 pos1과 pos2는 각각 앞선 pos에 대한 상대좌표로 정의된다. (pos1은 시작점 대비 상대좌표, pos2는 pos1 대비 상대좌표)
  - angle이 None일 경우 0으로 처리됩니다.
  - angle이 한 개만 입력된 경우, angle은 Circular path 상의 총 회전각을 적용합니다.
  - angle이 두 개가 입력된 경우, angle1은 circular path 상에서 정속으로 이동하는 총 회전각을, angle2는 가속과 감속을 위한 회전 구간의 회전각을 의미합니다. 이때 총 이동각은 angle1+2xangle2만큼 circular path 상을 움직입니다.
  - 옵션 ra 와 vel/acc에 따른 블렌딩 상태의 경로는 movej() 모션 설명을 참조하십시오.
  - app\_type이 DR\_MV\_APP\_WELD인 경우에 이동속도는 vel인자의 입력값이 아닌 app\_weld\_set\_weld\_cond()에 입력된 속도설정으로 대체됩니다.
  - ori(이동자세 선택)옵션은 아래와 같이 정의됩니다.
- DR\_MV\_ORI\_TEACH(교시자세) : 현재 자세로부터 포즈 2의 교시 자세까지 이동거리에 비례하여 자세를 변화하며 이동합니다. 포즈1의 교시자세는 무시됩니다.
  - DR\_MV\_ORI\_FIXED(고정자세) : 현재 자세를 포즈2 지점까지 유지한 채 경로를 따라 이동합니다.
  - DR\_MV\_ORI\_RADIAL(원주구속자세) : 현재 자세에서의 반경방향의 자세를 포즈2 지점까지 유지한 채 이동합니다



## 리턴

값	설명
0	성공
음수값	오류

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 #예제 1. x1으로 조인트모션 시작 후 2초 후에 D-Out
2
3 p0 = posj(-148,-33,-54,180,92,32)
4
5 movej(p0, v=30, a=30)
6
7 x1 = posx(784, 443, 770, 0, 180, 0)
8
9 amovejx(x1, vel=100, acc=200, sol=2)    # x1으로 조인트모션 및 즉시 다음명령 수행
10
11 wait(2)          # 2초간 프로그램 일시중지 (모션은 진행 중)
12
13 set_digital_output(1, 1)      # D-Out(1번채널) ON
14
15 mwait(0)

```

## 관련 명령어

- posx(x=0, y=0, z=0, A=0, B=0, C=0, ori\_type=None, sol=None, turn=None)(p. 30)
- set\_velx(vel1, vel2)(p. 48)
- (3.2.2-ko\_KR) set\_velx(vel)<sup>14</sup>
- set\_accx(acc1, acc2)(p. 50)
- (3.2.2-ko\_KR) set\_accx(acc)<sup>15</sup>

<sup>14</sup> [https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/202473924/3.2.2-ko\\_KR+set\\_velx+vel](https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/202473924/3.2.2-ko_KR+set_velx+vel)

<sup>15</sup> [https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/202179059/3.2.2-ko\\_KR+set\\_accx+acc](https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/202179059/3.2.2-ko_KR+set_accx+acc)

- [set\\_tcp\(name\)\(p. 52\)](#)
- [set\\_ref\\_coord\(coord\)\(p. 53\)](#)
- [mwait\(time=0\)\(p. 132\)](#)
- [movec\(\)\(p. 71\)](#)

### 3.4.5 amovesj()

#### 기능

비동기(async.)방식의 movesj() 모션으로 async 처리 외에는 movesj()와 동일하게 동작합니다.

amovesj()에 의한 모션이 종료되기 전에 발생하는 새로운 모션지령은 안전상의 이유로 오류를 발생시킵니다. 따라서 amovej()와 이어지는 모션명령어 사이에는 mwait() 또는 check\_motion() 등을 사용하여 amovesj() 모션 종료를 확인한 후 새로운 모션 명령어가 진행되도록 해야 합니다.

#### 비교

- movesj(pos\_list): 현재위치에서 출발하여 pos\_list의 끝점에 도달(정지)한 후에 다음 명령 수행합니다.
- amovesj(pos\_list): 현재위치에서 출발하여 pos\_list의 끝점 도달(정지)여부와 관계없이 즉시 다음 명령을 수행합니다.

#### 인수

인수명	자료형	기본값	설명
pos_list	list (posj)	-	posj list
vel (v)	float	None	velocity(모든 축에 동일) 또는 velocity(축별 velocity)
	list (float[6])		
acc (a)	float	None	acceleration(모든 축에 동일) 또는 acceleration(축별 acceleration)
	list (float[6])		
time (t)	float	None	도달 시간 [sec]

인수명	자료형	기본값	설명
mod	int	DR_MV_MOD_ABS	<p>이동 기준</p> <ul style="list-style-type: none"> <li>• DR_MV_MOD_ABS: 절대</li> <li>• DR_MV_MOD_REL: 상대</li> </ul>

### ❶ 알아두기

- 단축 인수를 지원합니다. (v:vel, a:acc, t:time)
- vel이 None인 경우 \_global\_velj가 적용됩니다. (\_global\_velj 초기값은 0.0이며, set\_velj에 의해 설정 가능)
- acc이 None인 경우 \_global\_accj가 적용됩니다. (\_global\_accj 초기값은 0.0이며, set\_accj에 의해 설정 가능)
- time을 지정할 경우 vel, acc를 무시하고 time 기준으로 처리됩니다.
- time이 None인 경우 0으로 처리됩니다.
- mod가 DR\_MV\_MOD\_REL인 경우 pos\_list의 각 pos는 앞선 pos에 대한 상대좌표로 정의됩니다. (pos\_list=[q1, q2, ..., q(n-1), q(n)]로 이루어질 때 q1은 시작점 대비 상대각도, q(n)은 q(n-1) 대비 상대좌표)
- 선행모션과 후행모션에 대한 온라인 블랜딩을 지원하지 않습니다.

### 리턴

값	설명
0	성공
음수값	오류

### 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시

예외	설명
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1   # 예제 1. q1~q5 경유하는 스플라인모션 시작 후 3초 후에 D-Out
2
3   q0 = posj(0,0,0,0,0,0)
4
5   movej(q0, vel=30, acc=60)           # 초기위치(q0)로 joint모션 이동
6
7   q1 = posj(10, -10, 20, -30, 10, 20)      # posj 변수(관절각) q1 정
8   의
9
10  q2 = posj(25, 0, 10, -50, 20, 40)
11
12  q3 = posj(50, 50, 50, 50, 50, 50)
13
14  q4 = posj(30, 10, 30, -20, 10, 60)
15
16
17
18
19  qlist = [q1, q2, q3, q4, q5]
20
21  # q1~q5를 경유점 집합으로 하는 리스트(qlist) 정의
22
23
24
25  amovesj(qlist, vel=30, acc=100)
26
27  # qlist에 정의된 경유점 집합을 연결하는 스플라인 곡선을 최대속도
28
29  # 30(mm/sec), 최대가속도 100(mm/sec2)로 움직임, 모션시작 즉시
30
31  # 다음명령 수행
32
33  wait(3)                         # 3초간 프로그램 일시중지 (모션은
34  진행 중)
35  set_digital_output(1, 1)         # D-Out(1번채널) ON
36
37  mwait(0)                        # 모션이 종료할 때까지 프로그램 일시중지

```

## 관련 명령어

- [posj\(J1=0, J2=0, J3=0, J4=0, J5=0, J6=0\)\(p. 29\)](#)
- [set\\_velj\(vel\)\(p. 45\)](#)
- [set\\_accj\(acc\)\(p. 47\)](#)
- [mwait\(time=0\)\(p. 132\)](#)
- [movesj\(\)\(p. 77\)](#)

### 3.4.6 amovesx()

#### 기능

비동기(async.)방식의 movesx 모션으로 async 처리 외에는 movesx()와 동일하게 동작합니다.

amovesx()에 의한 모션이 종료되기 전에 발생하는 새로운 모션지령은 안전상의 이유로 오류를 발생시킵니다. 따라서 amovesx()와 이어지는 모션명령어 사이에는 mwait() 또는 check\_motion() 등을 사용하여 amovesx() 모션이 종료를 확인한 후 새로운 모션 명령어가 진행되어야 합니다.

#### 비교

- movesx(pos\_list): 현재위치에서 출발하여 pos\_list의 끝점에 도달(정지)한 후에 다음 명령 수행
- amovesx(pos\_list): 현재위치에서 출발하여 pos\_list의 끝점 도달(정지)여부와 관계없이 즉시 다음 명령 수행

#### 인수

인수명	자료형	기본값	설명
pos_list	list (posx)	-	posx list
vel (v)	float	None	velocity 또는 velocity1, velocity2
	list (float[2])		
acc (a)	float	None	acceleration 또는 acceleration1, acceleration2
	list (float[2])		
time (t)	float	None	도달 시간 [sec]

인수명	자료형	기본값	설명
ref	int	None	reference coordinate • DR_BASE: base coordinate • DR_WORLD: world coordinate • DR_TOOL: tool coordinate • user coordinate: 사용자 정의
mod	int	DR_MV_MOD_ABS	이동 기준 • DR_MV_MOD_ABS: 절대 • DR_MV_MOD_REL: 상대
vel_opt	int	DR_MVS_VEL_NONE	속도 옵션 • DR_MVS_VEL_NONE: 없음 • DR_MVS_VEL_CONST: 등속

### ❶ 알아두기

- 단축 인수를 지원합니다. (v:vel, a:acc, t:time)
- vel이 None인 경우 \_global\_velx가 적용됩니다. (\_global\_velx 초기값은 0.0이며, set\_velx에 의해 설정 가능)
- acc이 None인 경우 \_global\_accx가 적용됩니다. (\_global\_accx 초기값은 0.0이며, set\_accx에 의해 설정 가능)
- vel에 하나의 인자를 입력한 경우(예를 들어, vel=30) 입력된 인자는 모션의 선속도에 대응되며, 각속도는 선속도에 비례하여 결정됩니다.
- acc에 하나의 인자를 입력한 경우(예를 들어, acc=60) 입력된 인자는 모션의 선가속도에 대응되며, 각가속도는 선가속도에 비례하여 결정됩니다.
- time을 지정할 경우 vel, acc를 무시하고 time 기준으로 처리됩니다.
- time이 None인 경우 0으로 처리됩니다.
- ref가 None인 경우, \_g\_coord가 적용됩니다. (\_g\_coord 초기값은 DR\_BASE이며, set\_ref\_coord 명령에 의해 설정 가능)
- mod가 DR\_MV\_MOD\_REL인 경우 pos\_list의 각 pos는 앞선 pos에 대한 상대좌표로 정의됩니다. (pos\_list=[p1, p2, ..., p(n-1), p(n)]로 이루어질 때 p1은 시작점 대비 상대각도, p(n)은 p(n-1) 대비 상대좌표)
- 선행모션과 후행모션에 대한 온라인 블랜딩을 지원하지 않습니다.

### ⚠ 주의

vel\_opt=DR\_MVS\_VEL\_CONST 옵션(등속모션) 선택 시, 입력된 경유점 간 거리 및 속도조건에 따라 등속모션이 불가능할 수 있으며 이 경우에 변속모션 (vel\_opt=DR\_MVS\_VEL\_NONE)으로 자동 전환됩니다.

## 리턴

값	설명
0	성공
음수값	오류

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 #예제 1. x1~x6 경유하는 스플라인모션 시작 후 3초 후에 D-Out >
2
3 P0 = posj(0,0,90,0,90,0)
4
5 movej(P0)
6
7 x0 = posx(600, 43, 500, 0, 180, 0)           # posx 변수(공간좌표/자세) x0 정의
8
9 movel(x0, vel=100, acc=200) # 초기위치 x0로 line모션
10
11 x1 = posx(600, 600, 600, 0, 175, 0)          # posx 변수(공간좌표/자세) x1 정의
12
13 x2 = posx(600, 750, 600, 0, 175, 0)
14
15 x3 = posx(150, 600, 450, 0, 175, 0)
16
17 x4 = posx(-300, 300, 300, 0, 175, 0)
18

```

```

19   x5 = posx(-200, 700, 500, 0, 175, 0)
20
21   x6 = posx(600, 600, 400, 0, 175, 0)
22
23
24
25   xlist = [x1, x2, x3, x4, x5, x6] # x1~x6를 경유점 집합으로 하는 리스트 xlist 정의
26
27
28
29   amovesx(xlist, vel=[100, 30], acc=[200, 60], vel_opt=DR_MVS_VEL_NONE)
30
31   # 현재위치에서 시작하여 xlist에 정의된 경유점 집합을 연결하는 스플라인
32
33   # 곡선을 최대속도 100, 30(mm/sec, deg/sec), 최대가속도 200(mm/sec2),
34
35   # 60(deg/sec2)로 움직임, 모션시작 즉시 다음명령 수행
36
37   wait(3)    # 3초간 프로그램 일시중지 (모션은 진행 중)
38
39   set_digital_output(1, 1) # D-Out(1번채널) ON
40
41   mwait(0)  # 모션이 종료할 때까지 프로그램 일시중지

```

## 관련 명령어

- posx(x=0, y=0, z=0, A=0, B=0, C=0, ori\_type=None, sol=None, turn=None)(p. 30)
- set\_velx(vel1, vel2)(p. 48)
- (3.2.2-ko\_KR) set\_velx(vel)<sup>16</sup>
- set\_accx(acc1, acc2)(p. 50)
- (3.2.2-ko\_KR) set\_accx(acc)<sup>17</sup>
- set\_tcp(name)(p. 52)
- set\_ref\_coord(coord)(p. 53)
- mwait(time=0)(p. 132)
- movesx()(p. 81)

### 3.4.7 amoveb()

#### 기능

비동기(async.)방식의 moveb모션으로 async 처리 외에는 moveb()와 동일하게 동작하며 명령어를 수행한 후 바로 다음 라인을 수행합니다.

<sup>16</sup> [https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/202473924/3.2.2-ko\\_KR+set\\_velx+vel](https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/202473924/3.2.2-ko_KR+set_velx+vel)

<sup>17</sup> [https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/202179059/3.2.2-ko\\_KR+set\\_accx+acc](https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/202179059/3.2.2-ko_KR+set_accx+acc)

amoveb()에 의한 모션이 종료되기 전에 발생하는 새로운 모션지령은 안전상의 이유로 오류를 발생시킵니다. 따라서 amoveb()와 이어지는 모션명령어 사이에는 mwait() 또는 check\_motion() 등을 사용하여 amoveb() 모션 종료를 확인한 후 새로운 모션 명령어가 진행되도록 해야합니다.

### 비교

- moveb(seg\_list): 현재위치에서 출발하여 seg\_list의 끝점에 도달(정지)한 후에 다음 명령 수행
- amoveb(seg\_list): 현재위치에서 출발하여 seg\_list의 끝점 도달(정지)여부와 관계없이 즉시 다음 명령 수행

## 인수

인수명	자료형	기본값	설명
pos_list	list (posb)	-	posb list
vel (v)	float	None	velocity 또는 velocity1, velocity2
	list (float[2])		
acc (a)	float	None	acceleration 또는 acceleration1, acceleration2
	list (float[2])		
time (t)	float	None	도달 시간 [sec] <ul style="list-style-type: none"> <li>time 지정 시, vel, acc를 무시하고 time 기준으로 처리</li> </ul>
ref	int	None	reference coordinate <ul style="list-style-type: none"> <li>DR_BASE: base coordinate</li> <li>DR_WORLD: world coordinate</li> <li>DR_TOOL: tool coordinate</li> <li>user coordinate: 사용자 정의</li> </ul>
mod	int	DR_MV_MOD_ABS	이동 기준 <ul style="list-style-type: none"> <li>DR_MV_MOD_ABS: 절대</li> <li>DR_MV_MOD_REL: 상대</li> </ul>

인수명	자료형	기본값	설명
app_type	int	DR_MV_APP_NONE	<p>어플리케이션 연동 옵션</p> <ul style="list-style-type: none"> <li>• DR_MV_APP_NONE: 연동없음</li> <li>• DR_MV_APP_WELD: 용접연동</li> </ul>

### ❶ 알아두기

- 단축 인수를 지원합니다. (v:vel, a:acc, t:time)
- posb\_list는 최대 50개까지 입력할 수 있습니다.
- vel이 None인 경우 \_global\_velx가 적용됩니다. (\_global\_velx 초기값은 0.0이며, set\_velx에 의해 설정 가능)
- acc이 None인 경우, \_global\_accx가 적용됩니다. (\_global\_accx 초기값은 0.0이며, set\_accx에 의해 설정 가능)
- vel에 하나의 인자를 입력한 경우(예를들어, vel=30) 입력된 인자는 모션의 선속도에 대응되며, 각속도는 선속도에 비례하여 결정됩니다.
- acc에 하나의 인자를 입력한 경우(예를들어, acc=60) 입력된 인자는 모션의 선가속도에 대응되며, 각가속도는 선가속도에 비례하여 결정됩니다.
- time을 지정할 경우 vel, acc를 무시하고 time 기준으로 처리됩니다.
- time이 None인 경우 0으로 처리됩니다.
- ref가 None인 경우 \_g\_coord가 적용됩니다. (\_g\_coord 초기값은 DR\_BASE이며, set\_ref\_coord 명령에 의해 설정 가능)
- mod가 DR\_MV\_MOD\_REL인 경우 posb\_list의 각 pos는 앞선 pos에 대한 상대좌표로 정의됩니다.
- 선행모션과 후행모션에 대한 온라인 블렌딩은 지원하지 않습니다.
- app\_type이 DR\_MV\_APP\_WELD인 경우에 이동속도는 vel인자의 입력값이 아닌 app\_weld\_set\_weld\_cond()에 입력된 속도설정으로 대체됩니다.

### ⚠ 주의

- posb에서 blending radius가 0인 경우, 사용자 입력 오류가 나타납니다.
- 연속된 Line-Line segment가 같은 방향을 가질 경우 Line의 중복입력으로 사용자 입력 오류가 나타납니다.
- 블렌딩 구간에서 조건에 따라 급격하게 방향전환이 발생하게 되는 경우 급가속을 방지하기 위해 사용자 입력오류가 나타납니다.
- 선행모션과 후행모션에 대한 온라인 블렌딩은 지원하지 않습니다.

## 리턴

값	설명
0	성공
음수값	오류

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 # 예제 1. seg11~seg16의 경로를 따르는 모션 시작 후 3초 후에 D-Out
2
3 # Init Pose @ Jx1
4
5 Jx1 = posj(45,0,90,0,90,45)                      #초기 Joint위치
6
7 X0 = posx(370, 420, 650, 0, 180, 0)             #초기 Task위치
8
9
10
11 # CASE 1) ABSOLUTE
12
13 # Absolute Goal Poses
14
15 X1 = posx(370, 670, 650, 0, 180, 0)
16
17 X1a = posx(370, 670, 400, 0, 180, 0)
18

```

```

19 X1a2= posx(370, 545, 400, 0, 180, 0)
20
21 X1b = posx(370, 595, 400, 0, 180, 0)
22
23 X1b2= posx(370, 670, 400, 0, 180, 0)
24
25 X1c = posx(370, 420, 150, 0, 180, 0)
26
27 X1c2= posx(370, 545, 150, 0, 180, 0)
28
29 X1d = posx(370, 670, 275, 0, 180, 0)
30
31 X1d2= posx(370, 795, 150, 0, 180, 0)
32
33
34
35 seg11 = posb(DR_LINE, X1, radius=20)
36
37 seg12 = posb(DR_CIRCLE, X1a, X1a2, radius=20)
38
39 seg14 = posb(DR_LINE, X1b2, radius=20)
40
41 seg15 = posb(DR_CIRCLE, X1c, X1c2, radius=20)
42
43 seg16 = posb(DR_CIRCLE, X1d, X1d2, radius=20)
44
45 b_list1 = [seg11, seg12, seg14, seg15, seg16]
46
47 # 마지막경유점(seg16)의 blending radius는 무시됨
48
49 movej(Jx1, vel=30, acc=60, mod=DR_MV_MOD_ABS)
50
51 # 초기각도(Jx1)로 Joint모션
52
53 movel(X0, vel=150, acc=250, ref=DR_BASE, mod=DR_MV_MOD_ABS)
54
55 #초기위치(X0)로 line모션
56
57 amoveb(b_list1, vel=150, acc=250, ref=DR_BASE, mod=DR_MV_MOD_ABS)
58
59 # 현재위치에서 시작하여 seg11(LINE), seg12(CIRCLE), seg14(LINE),
60
61 # seg15(CIRCLE), seg16(CIRCLE)으로 이루어진 궤적을 속도 150(mm/sec)를
62
63 # 유지하며(가감속구간 제외) 움직임 (최종point는 X1d2).
64
65 # 각 segment의 끝점(X1, X1a2, X1b2, X1c2, X1d2)에서 20mm 거리에 도달하면
66
67 # 다음 segment로 blending이 시작됨
68
69 wait(3)                                     # 3초간 프로그램 일시중지 (모션은
70 진행 중)
71 set_digital_output(1 , 1)                  # D-Out(1번채널) ON

```

72

73

`mwait(0)`

# 모션이 종료할 때까지 프로그램 일시중지

## 관련 명령어

- `posb(seg_type, posx1, posx2=None, radius=0)`(p. 35)
- `set_velx(vel1, vel2)`(p. 48)
- `(3.2.2-ko_KR) set_velx(vel)`<sup>18</sup>
- `set_accx(acc1, acc2)`(p. 50)
- `(3.2.2-ko_KR) set_accx(acc)`<sup>19</sup>
- `set_tcp(name)`(p. 52)
- `set_ref_coord(coord)`(p. 53)
- `mwait(time=0)`(p. 132)
- `moveb()`(p. 85)

## 3.4.8 amove\_spiral()

### 기능

비동기(async.)방식의 `move_spiral`모션으로 async 처리 외에는 `move_spiral()`와 동일하게 동작하며 명령어를 수행한 후 바로 다음 라인을 수행합니다.

`amove_spiral()`에 의한 모션이 종료되기 전에 발생하는 새로운 모션지령은 안전상의 이유로 오류를 발생시킵니다. 따라서 `amove_spiral()`과 이어지는 모션명령어 사이에는 `mwait()` 또는 `check_motion()` 등을 사용하여 `amove_spiral()` 모션 종료를 확인한 후 새로운 모션명령이 진행되도록 해야합니다.

지정한 좌표계(ref) 상에서 현재위치를 중심으로 지정축(axis)에 수직한 평면 위의 나선궤적을 따르는 모션을 수행합니다. 이동거리(lmax)를 추가로 입력하면 원뿔의 꼭지점으로부터 시작하여 주위를 따라 도는 궤적의 모션을 수행합니다.

### ● 알아두기

- `move_spiral`: 현재위치에서 출발하여 spiral궤적의 끝에 도달(정지)한 후에 다음 명령 수행
- `amove_spiral`: 현재위치에서 출발하여 spiral궤적의 끝에 도달(정지)여부와 관계없이 즉시 다음 명령 수행

<sup>18</sup> [https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/202473924/3.2.2-ko\\_KR+set\\_velx+vel](https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/202473924/3.2.2-ko_KR+set_velx+vel)

<sup>19</sup> [https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/202179059/3.2.2-ko\\_KR+set\\_accx+acc](https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/202179059/3.2.2-ko_KR+set_accx+acc)

## 인수

인수명	자료형	기본값	범위	설명
rev	float	10	rev > 0	총 회전수 [revolution]
rmax	float	10	rmax > 0	spiral 최종 반경 [mm]
lmax	float	0		axis 방향으로 이동하는 거리 [mm]
vel (v)	float	None		velocity
acc (a)	float	None		acceleration
time (t)	float	None	time ≥ 0	총 수행시간 <sec>
axis	int	DR_AXIS_Z	-	axis <ul style="list-style-type: none"> <li>• DR_AXIS_X: x축</li> <li>• DR_AXIS_Y: y축</li> <li>• DR_AXIS_Z: z축</li> </ul>
ref	Int	DR_TOOL	-	reference coordinate <ul style="list-style-type: none"> <li>• DR_BASE : base coordinate</li> <li>• DR_WORLD : world coordinate</li> <li>• DR_TOOL : tool coordinate</li> <li>• user coordinate : 사용자 정의</li> </ul>

### ● 알아두기

- 단축 인수를 지원합니다. (v:vel, a:acc, t:time)
- rev는 spiral 모션의 총 회전수를 의미합니다.
- rmax는 spiral 모션의 최대 반경을 의미합니다.
- lmax는 모션 동안 axis 방향으로 병진하는 거리를 의미합니다. 단, 음수인 경우 -axis 방향 병진합니다.
- vel은 spiral 모션의 이동 속도를 의미합니다.
- vel이 None인 경우, \_global\_velx의 첫째 값(병진 속도)이 적용됩니다. (\_global\_velx 초기값은 0.0이며, set\_velx에 의해 설정 가능)
- acc는 spiral 모션의 이동 가속도를 의미합니다.

- acc가 None인 경우, \_global\_accx 첫째 값(병진 가속도)이 적용됩니다. (\_global\_accx 초기값은 0.0이며, set\_accx에 의해 설정 가능)
- time을 지정할 경우 vel, acc를 무시하고 time 기준으로 처리됩니다.
- time이 None인 경우 0으로 처리됩니다.
- axis는 Spiral 모션이 정의하는 평면에 수직인 축을 정의합니다.
- ref는 spiral 모션이 정의하는 기준 좌표계를 의미합니다.
- 선행모션과 후행모션에 대한 온라인 블렌딩은 지원하지 않습니다.

### 주의

- 경로 생성 시 Spiral 경로에 의한 회전각 가속도를 연산하여 값이 큰 경우 안정적인 모션을 위하여 에러가 발생나타날 수 있습니다.  
이 경우 vel, acc 또는 time 값을 작게 조정하는 것을 권장합니다.

### 리턴

값	설명
0	성공
음수값	오류

### 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

### 예제

```

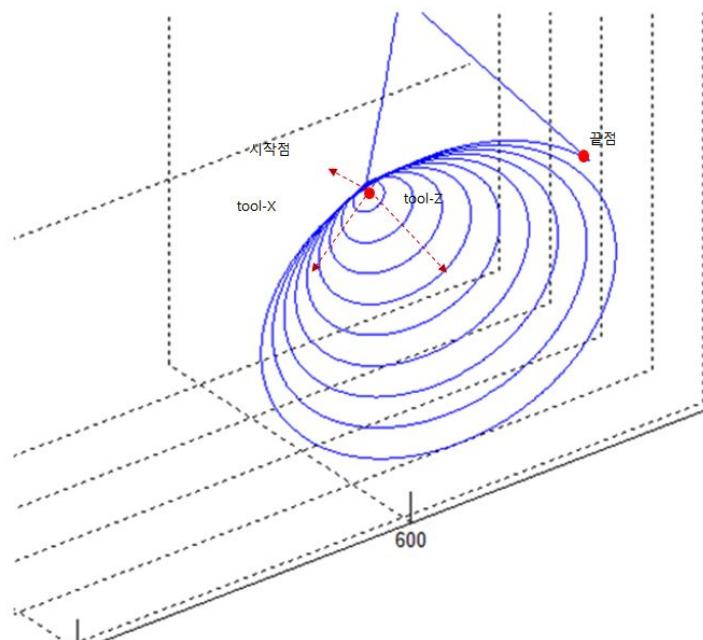
1 ## hole search
2

```

```

3   # (초기위치로 부터 Tool-Z방향의 회전중심으로 Tool-X/Y평면에서 0에서
4
5   # 30mm반경(rmax) 으로 9.5회전(revmax)을 하며 동시에 Tool-Z방향으로 50mm(lmax)
6
7   # 이동하는 spiral 궤적을 20초에 완료하는 모션, 모션시작 후 3초 후에
8
9   # D-Out(1번채널)
10
11
12
13 J00 = posj(0,0,90,0,60,0)
14
15 movej(J00,vel=30,acc=30)           #초기자세로 Joint이동
16
17 amove_spiral(rev=9.5,rmax=30.0,lmax=50.0,time=20.0, axis=DR_AXIS_Z, ref=DR_TOOL)
18
19 wait(3)
20
21 set_digital_output(1, 1) # D-Out(1번채널) ON
22
23 mwait(0)    #모션이 멈출때까지 대기

```



## 관련 명령어

- set\_velx(vel1, vel2)(p. 48)

- (3.2.2-ko\_KR) set\_velx(vel)<sup>20</sup>
- set\_accx(acc1, acc2)(p. 50)
- (3.2.2-ko\_KR) set\_accx(acc)<sup>21</sup>
- set\_tcp(name)(p. 52)
- set\_ref\_coord(coord)(p. 53)
- mwait(time=0)(p. 132)
- move\_spiral()(p. 90)

### 3.4.9 amove\_periodic()

#### 기능

비동기(async.)방식의 move\_periodic모션으로 async 처리 외에 move\_periodic()와 동일하게 동작하며 명령어를 수행한 후 바로 다음 라인을 수행합니다.

amove\_periodic()에 의한 모션이 종료되기 전에 발생하는 새로운 모션지령은 안전상의 이유로 오류를 발생시킵니다. 따라서 amove\_periodic()과 이어지는 모션명령어 사이에는 mwait() 또는 check\_motion() 등을 사용하여 amove\_periodic() 모션 종료를 확인한 후 새로운 모션명령이 진행되도록 해야합니다.

이 명령어는 현재위치에서 시작하는 상대모션어로 입력된 기준 좌표계(ref)의 각 축(병진 및 회전)에 대한 Sine함수 기반으로 주기모션을 수행합니다. 각 axis별 모션의 특성은 amp(amplitude)와 주기에 의해 결정되고 가감속 시간과 총 모션 시간은 주기, 반복 및 횟수에 의해 설정됩니다.

#### ❶ 알아두기

- move\_periodic: 현재위치에서 출발하여 periodic 궤적의 끝에 도달(정지)한 후에 다음 명령 수행
- amove\_periodic: 현재위치에서 출발하여 periodic 궤적의 끝에 도달(정지)여부와 관계없이 즉시 다음 명령 수행

#### 인수

인수명	자료형	기본값	범위	설명
amp	list (float[6])	-	0≤amp	Amplitude(-amp에서 +amp사이 모션) [mm] or [deg]
period	float or list (float[6])		0≤period	period(1주기 소요 시간)[sec]
atime	float	0.0	0≤atime	Acc-, dec- time [sec]

<sup>20</sup> [https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/202473924/3.2.2-ko\\_KR+set\\_velx+vel](https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/202473924/3.2.2-ko_KR+set_velx+vel)

<sup>21</sup> [https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/202179059/3.2.2-ko\\_KR+set\\_accx+acc](https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/202179059/3.2.2-ko_KR+set_accx+acc)

인수명	자료형	기본값	범위	설명
repeat	int	1	> 0	반복 횟수
ref	int	DR_TOOL	-	reference coordinate <ul style="list-style-type: none"> <li>• DR_BASE : base coordinate</li> <li>• DR_WORLD : world coordinate</li> <li>• DR_TOOL : tool coordinate</li> <li>• user coordinate : 사용자 정의</li> </ul>

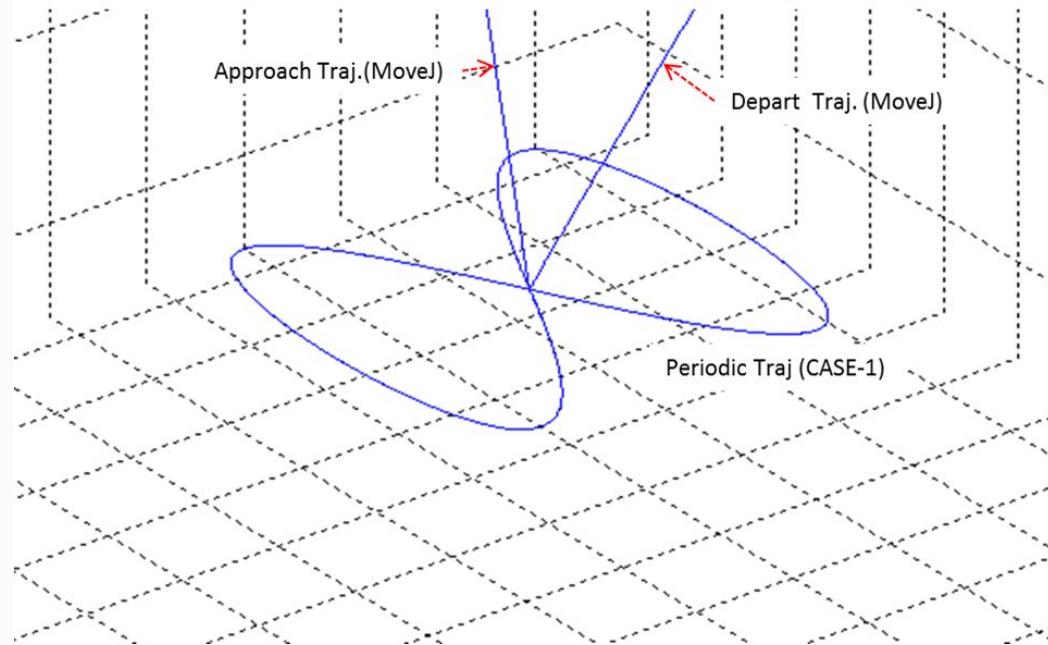
### ❶ 알아두기

- amp는 진폭(amplitude)을 의미하며, 각 축(x, y, z, rx, ry, rz) 별로 amp를 값으로 하는 6개 원소의 list 형태로 입력해야 합니다. 단, 주기 모션을 진행하지 않는 축 방향은 amp를 0으로 입력해야 합니다.
- period는 해당 방향 모션의 1회 반복 시간을 의미하며, 각 축(x, y, z, rx, ry, rz) 별 period를 값으로 하는 총 6개 원소의 list 형태로 입력하거나 대표값을 입력해야합니다.
- atime은 주기모션의 시작과 끝의 가속 및 감속 시간을 의미합니다. 입력된 가감속시간과 최대주기 \*1/4 중 큰 값이 적용됩니다. 입력된 가감속 시간이 전체모션시간의 1/2을 초과하는 경우 에러가 발생합니다.
- repeat은 가장 큰 period 값을 가지는 축(기준 축)의 반복 횟수를 정의하며, 이에 따라 총 모션 시간이 결정됩니다. 나머지 축의 반복 횟수는 모션 시간에 따라 자동 결정됩니다.
- 모션이 정상 종료되는 경우 종료 위치가 시작 위치와 일치하게 하도록 나머지 축 모션은 기준 축 모션이 종료되기 전에 먼저 종료될 수 있습니다. 모든 축의 모션이 동시에 종료되지 않는 경우 감속구간

에서의 경로는 이전 경로에서 벗어나게 됩니다. 관련한 사항은 아래 이미지를 참조하십시오

**CASE-1) All-axis motions end at the same time**

move\_periodic(amp=[100,100,0,0,0,0], period=[3.2,1.6,0,0,0,0], atime=3.1, repeat=2, ref=DR\_BASE)



- ref는 반복 모션의 기준 좌표계를 의미합니다.
- 모션명령 수행 시 최대속도 에러가 발생하는 경우 다음의 식을 참조하여 진폭 및 주기를 조정할 것을 제안합니다.  
**최대속도=진폭(amp)\*2\*pi(3.14)/주기(period)**  
(예, 진폭=10mm, 주기=1초인 경우 최대속도=62.83mm/sec)
- 선행모션과 후행모션에 대한 온라인 블렌딩은 지원하지 않습니다.

## 리턴

값	설명
0	성공
음수값	오류

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시

예외	설명
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 P0 = posj(0,0,90,0,90,0)
2
3 movej(P0)
4
5 amove_periodic(amp =[10,0,0,0,0.5,0], period=1, atime=0.5, repeat=5, ref=D
R_TOOL)
6
7 wait(1)
8
9 set_digital_output(1, 1)
10
11 mwait(0)
12
13 # Tool좌표계 x축(10mm진폭, 1초 주기)모션과 y회전축(진폭 0.5deg, 1초 주기)
14
15 # 모션을 총 5회 반복 수행
16
17 # periodic 모션을 시작하고 1초 후에 Digital_Output 채널1번을 SET(1) 한다.

```

## 관련 명령어

- [set\\_ref\\_coord\(coord\)\(p. 53\)](#)
- [move\\_periodic\(\)\(p. 94\)](#)

## 3.5 추가 기능

### 3.5.1 mwait(time=0)

#### 기능

선행된 모션 명령어와 다음 라인의 모션 명령어 사이의 대기 시간을 설정합니다. 대기 시간은 time[sec]에 입력한 시간에 따라 달라집니다.

#### 인수

인수명	자료형	기본값	설명
time	float	0	모션 끝난 후 대기 시간 [sec]

#### 리턴

값	설명
0	성공
음수값	오류

#### 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 #예제 1. q0로 모션시작 후 3초 후에 q1으로 움직여 모션정지까지 대기하였다가 q99로 이동
2
3 q0 = posj(0, 0, 90, 0, 90, 0)
4
5 amovej (q0, vel=10, acc=20)      # q0로 모션 및 즉시 다음 명령 수행
6
7 wait(3)                         # 3초간 프로그램 일시 중지(모션은
8 진행 중)
9
10 q1 = posj(0, 0, 0, 0, 90, 0)
11 amovej (q1, vel=10, acc=20)
12
13 # q0 모션을 유지(ra 인자 생략 시 DUPLICATE blending)하며 q1으로 중첩
14
15 # blending하는 모션 및 즉시 다음 명령 수행
16
17 mwait(0)                         # 모션이 종료할 때까지 프로그램 일시 중지
18
19 q99 = posj(0, 0, 0, 0, 0, 0)
20
21 movej (q99, vel=10, acc=20)      # q99으로 조인트 모션

```

## 관련 명령어

- [wait\(time\)](#)(p. 293)
- [amovej\(\)](#)(p. 100)
- [amovel\(\)](#)(p. 103)
- [amovejx\(\)](#)(p. 106)
- [amovec\(\)](#)(p. 109)
- [amovesj\(\)](#)(p. 113)
- [amovesx\(\)](#)(p. 116)
- [amoveb\(\)](#)(p. 119)
- [amove\\_spiral\(\)](#)(p. 124)
- [amove\\_periodic\(\)](#)(p. 128)

### 3.5.2 begin\_blend(radius=0)

#### 기능

블렌딩 구간을 시작합니다. 이후의 블렌딩 구간 인자 radius를 갖는 sync motion 명령어(movej, movel, movec, movejx)들은 기본적으로 인자로 설정된 radius를 이용해 blending됩니다. radius가 0인 경우 실질적인 blending 효

과가 없습니다. 또한 설정된 radius와 다른 blending radius의 설정이 필요한 경우는 해당 모션의 인자에 blending radius를 지정함으로써 예외적으로 blending radius의 변경이 가능합니다.

## 인수

인수명	자료형	기본값	설명
radius	float	0	Blending 시 radius

## 리턴

값	설명
0	성공

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시

## 예제

```

1 #1
2
3 begin_blend(radius=30)
4
5 # 이후의 radius 옵션을 가지는 모션 명령은 blending 구간을 30mm로
6
7 # 일괄설정함
8
9 Q1 = posj(0,0,90,0,90,0)
10
11 Q2 = posj(0,0,0,0,90,0)
12
13 movej(Q1, vel=10, acc=20)
14
15 # Q1 관절각으로 이동하며 Q1의 공간 위치로부터 30mm 전역 거리가 될 때
16
17 # 다음 모션을 수행하도록 설정됨
18
19 movej(Q2, time=5)
20
21 # 직전모션을 유지하며(모션중첩) Blending하여 Q2 관절각으로 이동하며,
22

```

```

23 # Q2의 공간 위치로부터 30mm 전역 거리가 될 때 다음 모션을 수행하도록
24
25 # 설정됨
26
27 movej(Q1, v=30, a=60, r=200)
28
29 # 직전모션을 유지하며(모션중첩) Blending하여 Q1 관절각으로 이동하며,
30
31 # Q1의 공간 위치로부터 200mm 거리가 될 때 다음 모션을 수행하도록
32
33 # 설정(전역 설정 미적용)
34
35 movej(Q2, v=30, a=60, ra= DR_MV_RA_OVERRIDE)
36
37 # 직전모션을 즉시 종료하며 Blending하여 Q2 관절각으로 이동
38
39
40 end_blend()      # blending 구간 일괄설정 해제
41

```

## 관련 명령어

- [end\\_blend\(\)](#)(p. 135)
- [movej\(\)](#)(p. 57)
- [movel\(\)](#)(p. 62)
- [movejx\(\)](#)(p. 67)
- [movec\(\)](#)(p. 71)

### 3.5.3 end\_blend()

#### 기능

블렌딩 구간을 종료합니다. begin\_blend()로 시작된 블렌딩 구간의 효력이 정지됨을 의미합니다.

#### 리턴

값	설명
0	성공

#### 예제

```

1 #1
2
3 begin_blend(radius=30)
4

```

```

5   # 이후의 radius 옵션을 가지는 모션명령은 blending구간을 30mm로 일괄설정
6
7   Q1 = posj(0,0,90,0,90,0)
8
9   Q2 = posj(0,0,0,0,90,0)
10
11 movej(Q1, vel=10, acc=20)
12
13 # Q1관절각으로 이동시키며 Q1의 공간위치로부터 30mm의 전역거리가 될 때
14
15 # 다음 모션을 수행하도록 설정
16
17 movej(Q2, time=5)
18
19 # 직전모션을 유지하며(모션중첩) Blending하여 Q2관절각으로 이동,
20
21 # Q2의 공간위치로부터 30mm의 전역거리가 될 때 다음 모션을
22
23 # 수행하도록 설정
24
25 movej(Q1, v=30, a=60, r=200)
26
27 # 직전모션을 유지하며(모션중첩) Blending하여 Q1관절각으로 이동시키며
28
29 # Q1의 공간위치로부터 200mm의 거리가 될 때 다음 모션을 수행하도록
30
31 # 설정 (전역설정 미적용)
32
33 movej(Q2, v=30, a=60, ra= DR_MV_RA_OVERRIDE)
34
35 # 직전모션을 즉시 종료시키며 Blending하여 Q2관절각으로 이동
36
37 end_blend()      # blending구간 일괄설정 해제

```

## 관련 명령어

- begin\_blend(radius=0)(p. 133)
- movej()(p. 57)
- movel()(p. 62)
- movejx()(p. 67)
- movec()(p. 71)

### 3.5.4 check\_motion()

#### 기능

현재 진행 중인 모션의 상태를 확인합니다.

## 리턴

값	설명
0	DR_STATE_IDLE (수행중인 모션이 없음)
1	DR_STATE_INIT (모션 연산 중)
2	DR_STATE_BUSY (모션이 수행 중)

## 예외

예외	설명
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 #1. q0로의 비동기모션 중 모션이 감속을 시작하면 다음모션(q99) 명령 수행
2
3 q0 = posj(0, 0, 90, 0, 90, 0)
4
5 q99 = posj(0, 0, 0, 0, 0, 0)
6
7 amovej (q0, vel=10, acc=20) # q0로 모션 및 즉시 다음명령 수행
8
9 while True:
10
11     if check_motion() == 0: # 모션이 완료 된 경우
12
13         amovej (q99, vel=10, acc=20) # q99로 조인트 모션
14
15         break
16
17     if check_motion() == 2: # 모션 중인 경우
18
19         pass
20
21     mwait(0)      # 모션이 종료할 때까지 프로그램 일시중지

```

## 관련 명령어

- [movej\(\)\(p. 57\)](#)
- [movel\(\)\(p. 62\)](#)
- [movejx\(\)\(p. 67\)](#)
- [movec\(\)\(p. 71\)](#)
- [movesj\(\)\(p. 77\)](#)
- [movesx\(\)\(p. 81\)](#)
- [moveb\(\)\(p. 85\)](#)
- [move\\_spiral\(\)\(p. 90\)](#)
- [move\\_periodic\(\)\(p. 94\)](#)
- [amovej\(\)\(p. 100\)](#)
- [amovel\(\)\(p. 103\)](#)
- [amovejx\(\)\(p. 106\)](#)
- [amovec\(\)\(p. 109\)](#)
- [amovesj\(\)\(p. 113\)](#)
- [amovesx\(\)\(p. 116\)](#)
- [amoveb\(\)\(p. 119\)](#)
- [amove\\_spiral\(\)\(p. 124\)](#)
- [amove\\_periodic\(\)\(p. 128\)](#)

### 3.5.5 stop(st\_mode)

#### 기능

수행 중인 모션을 정지합니다. 인자로 받는 st\_mode에 따라 정지시간이 결정되며 진행중인 경로에서 벗어나지 않습니다.

본 명령어는 로봇의 동작을 정지시키기 위해 사용되며 프로그램의 실행을 중지시키는 기능은 수행되지 않습니다. 프로그램의 실행을 정지시키기 위해서는 exit() 명령을 추가로 실행시키십시오. 인자값 DR\_QSTOP\_STO, DR\_QSTOP은 각각 Functional Safety에서 정의하는 Stop Category 1(최대감속 후 토크off)과 2(최대감속)에 대응되나 정지 이후의 Torque off 등의 동작은 연계되지 않습니다. DR\_SSTOP의 경우 최대감속시간보다 약 1.5배정도 긴 감속시간동안 정지합니다. DR\_HOLD의 경우 감속구간없이 즉시 정지시킵니다.

## 인수

인수명	자료형	기본값	설명
st_mode	int	-	<p>stop mode</p> <ul style="list-style-type: none"> <li>• DR_QSTOP_STO: Quick stop (Stop Category 1 without STO(Safe Torque Off))</li> <li>• DR_QSTOP: Quick stop (Stop Category 2)</li> <li>• DR_SSTOP: Soft Stop</li> <li>• DR_HOLD: HOLD stop</li> </ul>

## 리턴

값	설명
0	성공
음수값	오류

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 #1. x1으로 이동 시작 2초 후에 Soft Stop으로 모션 종료
2
3 p0 = posj(-148,-33,-54,180,92,32)
4
5 movej(p0, v=30, a=30)

```

```

6
7   x1 = posx(784, 543, 570, 0, 180, 0)
8
9   amovel (x1, vel=100, acc=200) # x1으로 모션 및 즉시 다음 명령 수행
10
11  wait(2)    # 2초간 프로그램 일시 중지
12
13  stop(DR_SSTOP)    # Soft Stop하여 모션 정지

```

## 관련 명령어

- movej()(p. 57)
- movel()(p. 62)
- movejx()(p. 67)
- movec()(p. 71)
- movesj()(p. 77)
- movesx()(p. 81)
- moveb()(p. 85)
- move\_spiral()(p. 90)
- move\_periodic()(p. 94)
- amovej()(p. 100)
- amovel()(p. 103)
- amovejx()(p. 106)
- amovec()(p. 109)
- amovesj()(p. 113)
- amovesx()(p. 116)
- amoveb()(p. 119)
- amove\_spiral()(p. 124)
- amove\_periodic()(p. 128)

### 3.5.6 change\_operation\_speed(speed)

#### 기능

작동 속도를 조정합니다. 인자는 현재 설정된 속도에 대한 상대적인 비율을 백분율로 환산한 값으로 1에서 100까지의 값을 갖습니다. 따라서 50은 현재 설정된 속도의 50 Percent로 속도를 줄인다는 의미입니다.

#### 인수

인수명	자료형	기본값	설명
speed	int	-	operation speed(10~100)

## 리턴

값	설명
0	성공
음수값	오류

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1   change_operation_speed(10)
2
3   change_operation_speed(100)
4
5   #1. q0로 지정 속도 모션 및 지정 속도의 20%로 모션
6
7   q0 = posj(0, 0, 90, 0, 90, 0)
8
9   movej (q0, vel=10, acc=20)      # q0로 10mm/sec 속도로 이동
10
11  change_operation_speed(10)      # 이후 실행되는 모든 모션의 속도는 지정 속도의 10%
12
13  q1 = posj(0, 0, 0, 0, 90, 0)
14
15  movej (q1, vel=10, acc=20)      # q1으로 1mm/sec 속도(10mm/sec의 10%)로 이동
16
17  change_operation_speed(100)     # 이후 실행되는 모든 모션의 속도는 지정속도의 100%
18

```

19	movej (q0, vel=10, acc=20) # q0로 10mm/sec 속도로 이동(10mm/sec의 100%)
----	--

## 관련 명령어

- movej()(p. 57)
- movel()(p. 62)
- movejx()(p. 67)
- movec()(p. 71)
- movesj()(p. 77)
- movesx()(p. 81)
- moveb()(p. 85)
- move\_spiral()(p. 90)
- move\_periodic()(p. 94)
- amovej()(p. 100)
- amovel()(p. 103)
- amovejx()(p. 106)
- amovec()(p. 109)
- amovesj()(p. 113)
- amovesx()(p. 116)
- amoveb()(p. 119)
- amove\_spiral()(p. 124)
- amove\_periodic()(p. 128)

### 3.5.7 wait\_manual\_guide()

#### 기능

프로그램 수행 중 핸드가이딩(콕핏 또는 TP의 직접교시 버튼을 누르며 로봇의 위치변경)을 허용하며 사용자는 핸드 가이딩을 완료한 후 아래의 두 가지 방법 중 하나를 수행하면 다음 명령어로 진행합니다(프로그램이 종료되지 않는 한, 사용자가 자유롭게 핸드가이딩을 수행한 후 아래의 방법 중 하나를 실행할 때까지 본 명령어에서 대기합니다).

1. TP에 발생한 ‘핸드가이딩 수행’ 팝업창에서 ‘확인’ 또는 ‘완료’ 버튼을 누릅니다.
2. Safety I/O 설정을 통해 ‘manual guide 해제’로 지정한 Digital Input 채널에 신호를 인가합니다.

본 명령어를 정상적으로 실행하기 위해서는 현재의 TCP위치 및 핸드가이딩 중의 로봇의 TCP 위치가 협동작업영역 (collaborative workspace)내에 위치하여야 하므로 사전에 핸드가이딩을 수행할 영역을 협동작업영역으로 설정하고 활성화한 후에 실행하시기 바랍니다. 작업자의 안전을 위하여 현재의 위치 또는 핸드가이딩 중 협동작업영역을 벗어나는 경우에는 에러가 발생하며 작업 프로그램이 정지됩니다.

#### 리턴

값	설명
0	성공

값	설명
음수값	오류

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 # 프로그램 실행 전 협동 작업 영역 설정
2
3 # 면1: 점-1(1000,1000), 점-2(0,0)
4
5 # 면2: 점-1(1000,-1000), 점-2(0,0)
6
7 # 영역1 활성화 - 점(1000,0)
8
9
10 j00 = posj(0,0,90,0,90,0)
11
12 movej(j00,vel=20,acc=40) # 협동 작업 영역으로 진입
13
14 wait_manual_guide() # TP에 발생된 팝업의 '완료'버튼을 누를때까지 직접교시 가능
15
16 pos1 = get_current_posx() # 직접교시된 점을 pos1에 저장합니다.
17
18 dposa = posx(0,0,-100,0,0,0)
19
20 movel(dposa, vel=300, acc=600, ref=DR_TOOL)
21
22 # 교시된 위치로 부터 Tool-Z방향으로 100mm 후퇴합니다.
23

```

## 관련 명령어

- [movej\(\)\(p. 57\)](#)
- [movel\(\)\(p. 62\)](#)
- [movejx\(\)\(p. 67\)](#)
- [movec\(\)\(p. 71\)](#)
- [movesj\(\)\(p. 77\)](#)
- [movesx\(\)\(p. 81\)](#)
- [moveb\(\)\(p. 85\)](#)
- [move\\_spiral\(\)\(p. 90\)](#)
- [move\\_periodic\(\)\(p. 94\)](#)
- [amovej\(\)\(p. 100\)](#)
- [amovel\(\)\(p. 103\)](#)
- [amovejx\(\)\(p. 106\)](#)
- [amovec\(\)\(p. 109\)](#)
- [amovesj\(\)\(p. 113\)](#)
- [amovesx\(\)\(p. 116\)](#)
- [amoveb\(\)\(p. 119\)](#)
- [amove\\_spiral\(\)\(p. 124\)](#)
- [amove\\_periodic\(\)\(p. 128\)](#)

### 3.5.8 wait\_nudge()

#### 기능

프로그램 수행 중 일시정지상태에서 사용자의 넛지입력(로봇에 외력을 가하는 행동)을 통한 작업재개를 허용합니다. 로봇이 정지한 상태에서 설정으로 입력한 외력하한치(threshold force) 이상의 외력이 인가되면 설정된 재개시간 후에 다음 명령어로 진행되며, 이는 프로그램 진행 중의 인터락(interlock)으로 활용될 수 있습니다.

다만, 로봇의 관절형태가 특이점영역에 있는 경우, 또는 넛지입력 후 지속적으로 힘이인가되는 경우 안전을 위해 에러가 발생하는 것을 유의하십시오.

본 명령어를 정상적으로 실행하기 위해서는 현재의 TCP위치가 협동작업영역(collaborative workspace)내에 위치하여야 하므로 사전에 협동작업영역을 활성화하며 명령어를 수행할 로봇의 위치가 협동작업영역내에 있도록 설정하시기 바랍니다. 넛지입력을 인지할 외력의 크기 및 넛지입력 후 프로그램 재개시간의 설정 역시 협동작업영역 설정창에서 입력할 수 있습니다.

#### 리턴

값	설명
0	성공

값	설명
음수값	오류

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 # 프로그램 실행 전 협동 작업 영역 설정
2
3 # 면1: 점-1(1000,1000), 점-2(0,0)
4
5 # 면2: 점-1(1000,-1000), 점-2(0,0)
6
7 # 영역1 활성화 - 점(1000,0)
8
9
10 j00 = posj(0,0,90,0,90,0)
11
12 movej(j00,vel=20,acc=40) # 협동 작업 영역으로 진입
13
14 wait_nudge()           # 로봇에 nudge_threshold이상의 외력을 가할 때까지 대기
15
16 dposa = posx(0,0,-100,0,0,0)
17
18 movel(dposa, vel=300, acc=600, ref=DR_TOOL)
19
20 # 현재 위치로 부터 Tool-Z방향으로 100mm 후퇴합니다.
21

```

## 관련 명령어

- [movej\(\)\(p. 57\)](#)
- [movel\(\)\(p. 62\)](#)
- [movejx\(\)\(p. 67\)](#)
- [movec\(\)\(p. 71\)](#)
- [movesj\(\)\(p. 77\)](#)
- [movesx\(\)\(p. 81\)](#)
- [moveb\(\)\(p. 85\)](#)
- [move\\_spiral\(\)\(p. 90\)](#)
- [move\\_periodic\(\)\(p. 94\)](#)
- [amovej\(\)\(p. 100\)](#)
- [amovel\(\)\(p. 103\)](#)
- [amovejx\(\)\(p. 106\)](#)
- [amovec\(\)\(p. 109\)](#)
- [amovesj\(\)\(p. 113\)](#)
- [amovesx\(\)\(p. 116\)](#)
- [amoveb\(\)\(p. 119\)](#)
- [amove\\_spiral\(\)\(p. 124\)](#)
- [amove\\_periodic\(\)\(p. 128\)](#)

### 3.5.9 enable.Alter\_motion(n,mode,ref,limit\_dPOS,limit\_dPOS\_per)

#### 기능

경로 수정 기능을 활성화합니다. 경로 생성의 단위 주기는 100msec이며 입력 인자 n을 설정하여 경로 생성 주기 ( $n * 100\text{msec}$ )를 변경할 수 있습니다. 입력 인자 mode를 통해 alter\_motion()의 입력값의 의미를 2가지 모드(누적량 모드, 증분량 모드) 중 하나로 선택하여 사용할 수 있습니다. 누적량 모드의 경우 현재의 모션경로에 대한 절대적 증분위치/자세만큼 경로 수정량이 반영됩니다. 증분량 모드의 경우 바로 현재의 절대적 증분위치/자세에 입력된 증분 위치/자세만큼 경로 수정량이 추가되어 반영됩니다. 입력 인자 ref를 통해 기준 좌표계를 설정할 수 있습니다. 입력 인자 limit\_dPOS, limit\_dPOS\_per를 통해 각각 누적량, 증분량의 한계치를 설정할 수 있습니다. 한계치를 벗어나는 위치 값의 한계치에 수렴한 값으로 경로 수정량이 재조정됩니다.

#### 인수

인수명	자료형	기본값	설명
n	int	None	경로 생성 주기
mode	Int	None	경로 수정 모드 <ul style="list-style-type: none"> <li>• DR_DPOS : 누적량</li> <li>• DR_DVEL : 증분량</li> </ul>

인수명	자료형	기본값	설명
ref	int	None	<p>reference coordinate</p> <ul style="list-style-type: none"> <li>• DR_BASE: base coordinate</li> <li>• DR_WORLD: world coordinate</li> <li>• DR_TOOL: tool coordinate</li> <li>• user coordinate: 사용자 정의</li> </ul>
limit_dPOS	list(float[2])	None	<p>첫번째 값 : 이동량 제한 값[mm]      두번째 값 : 회전량 제한 값[deg]</p>
limit_dPOS_per	list(float[2])	None	<p>첫번째 값 : 이동량 제한 값[mm]      두번째 값 : 회전량 제한 값[deg]</p>

### ❶ 알아두기

- alter\_motion()은 사용자 thread 내에서만 동작합니다.
- ref가 None인 경우, \_g\_coord 적용(\_g\_coord 초기값은 DR\_BASE이며, set\_ref\_coord 명령에 의해 설정 가능)
- limit\_dPOS, limit\_dPOS\_per를 설정하지 않을 시 누적량, 증분량의 한계를 제한하지 않습니다.

### 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

### 예제

```

1  def alter_thread():
2
3      alter_motion(dx)

```

```

4
5
6 dX = [10,0,0,10,0,0]
7
8
9 J00 = posj(0,0,90,0,90)
10
11 X1 = posx(559.0, 200, 651.5, 180, -180.0, 180)
12
13 X2 = posx(559.0, 200, 400, 180, -180.0, 180)
14
15
16 movej(J00,vel=50,acc=100)
17
18
19 enable.Alter_motion(n=5,mode=DR_DPOS, ref=DR_BASE, limit_dPOS=[50,90],
20 limit_dPOS_per=[50,50])
21 # 경로 수정 기능 활성화
22
23 # 생성주기:(5*100)msec, 모드:누적량, 기준좌표계:베이스
24
25 # 누적량 제한:50mm,90deg, 증분량 제한:50mm, 50deg
26
27
28 th_id = thread_run(alter_thread, loop=True)
29
30
31 movel(X1,v=50,a=100,r=30)
32
33 movel(X2,v=50,a=100)
34
35
36 thread_stop(th_id)
37
38 disable.Alter_motion() # 경로 수정 기능 비활성화

```

## 관련 명령어

- alter\_motion(pos)(p. 148)
- disable.Alter\_motion()(p. 150)

### 3.5.10 alter\_motion(pos)

#### 기능

입력 인자 pos에 해당하는 양만큼 경로 수정을 진행합니다.



주의

- `alter_motion()`은 사용자 thread 내에서만 동작합니다.

### 알아두기

- `alter_motion()`은 `enable_alter_motion()`을 통해 경로보정기능이 활성화된 경우에만 유효합니다.
- `enable_alter_motion`의 설정 값 `limit_dPOS`, `limit_dPOS_per`에 따라 경로 수정량은 조정될 수 있습니다.
- `pos`의 방향값은 Fixed XYZ로 설정하여야 됩니다.

### 인수

인수명	자료형	기본값	설명
pos	list (float[6]) posx	-	position list posx

### 예외

예외	설명
<code>DR_Error</code> ( <code>DR_ERROR_TYPE</code> )	인수들의 데이터형 오류 시
<code>DR_Error</code> ( <code>DR_ERROR_VALUE</code> )	인수의 값이 유효하지 않을 시
<code>DR_Error</code> ( <code>DR_ERROR_RUNTIME</code> )	C Extension 모듈 에러 발생 시
<code>DR_Error</code> ( <code>DR_ERROR_STOP</code> )	프로그램 강제 종료 시

### 예제

```

1  def alter_thread():
2
3      alter_motion(dX)
4
5
6
7  dX = [10,0,0,10,0,0]
8
9

```

```

10
11 J00 = posj(0,0,90,0,90)
12
13 X1 = posx(559.0, 200, 651.5, 180, -180.0, 180)
14
15 X2 = posx(559.0, 200, 400, 180, -180.0, 180)
16
17
18
19 movej(J00,vel=50,acc=100)
20
21
22
23 enable.Alter_motion(n=10,mode=DR_DPOS, ref=DR_BASE, limit_dPOS=[50,90],
24 limit_dPOS_per=[10,10])
25 # 경로 수정 기능 활성화
26
27 # 생성주기:(5*100)msec, 모드:누적량, 기준좌표계:베이스
28
29 # 누적량 제한:50mm,50deg, 증분량 제한:10mm, 10deg
30
31
32
33 th_id = thread_run(alter_thread, loop=True)
34
35
36
37 movel(X1,v=50,a=100,r=30)
38
39 movel(X2,v=50,a=100)
40
41
42
43 thread_stop(th_id)
44
45 disable.Alter_motion() # 경로 수정 기능 비활성화

```

## 관련 명령어

- enable.Alter\_motion(n,mode,ref,limit\_dPOS,limit\_dPOS\_per)(p. 146)
- disable.Alter\_motion()(p. 150)

### 3.5.11 disable.Alter\_motion()

#### 기능

경로 수정 기능을 비활성화 합니다.

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1  def alter_thread():
2
3      alter_motion(dX)
4
5
6
7  dX = [10,0,0,10,0,0]
8
9
10 J00 = posj(0,0,90,0,90)
11
12 X1 = posx(559.0, 200, 651.5, 180, -180.0, 180)
13
14 X2 = posx(559.0, 200, 400, 180, -180.0, 180)
15
16
17
18 movej(J00,vel=50,acc=100)
19
20
21
22
23 enable_alter_motion(n=10,mode=DR_DPOS, ref=DR_BASE, limit_dPOS=[50,90],
24 limit_dPOS_per=[50,50])
25 # 경로 수정 기능 활성화
26
27 # 생성주기:(5*100)msec, 모드:누적량, 기준좌표계:베이스
28
29 # 누적량 제한:50mm,50deg, 증분량 제한:10mm, 10deg

```

```

30
31
32
33 th_id = thread_run(alter_thread, loop=True)
34
35
36
37 movel(X1,v=50,a=100,r=30)
38
39 movel(X2,v=50,a=100)
40
41
42
43 thread_stop(th_id)
44
45 disable.Alter_motion() # 경로 수정 기능 비활성화

```

## 관련 명령어

- enable.Alter\_motion(n, mode, ref, limit\_dPOS, limit\_dPOS\_per)(p. 146)
- alter\_motion(pos)(p. 148)

### 3.5.12 check\_robot\_mastering()

#### 기능

현재 로봇이 마스터링이 필요한 상태인지 아닌지 확인합니다.

**ⓘ 마스터링이 필요한 상태**

Servo On 상태에서 제어기 전원을 강제로 내렸을 때  
STO 에러가 발생한 후 다시 Servo On을 할 때

#### 인수

해당 사항 없음

#### 리턴

값	설명
0	마스터링 완료
1	마스터링 필요

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1   ret = check_robot_mastering()
2   if ret == 1:
3       move_home()

```

## 관련 명령어

- [move\\_home\(\)](#)(p. 99)

### 3.5.13 motion\_pause()

#### 기능

수행 중인 모션을 일시 정지합니다. 현재 수행 중인 모션만 일시 정지되며, 비동기로 작동하는 다른 기능에는 영향을 주지 않습니다.

#### 인수

없음

#### 리턴

값	설명
0	성공

값	설명
음수값	오류

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 amovej([0, 0, 90, 0, 90, 0], 60, 30)
2 wait(1)
3 motion_pause()
4 wait(1)
5 motion_resume()

```

## 관련 명령어

- [movej\(\)](#)(p. 57)
- [movel\(\)](#)(p. 62)
- [movejx\(\)](#)(p. 67)
- [movec\(\)](#)(p. 71)
- [movesj\(\)](#)(p. 77)
- [movesx\(\)](#)(p. 81)
- [moveb\(\)](#)(p. 85)
- [move\\_spiral\(\)](#)(p. 90)
- [move\\_periodic\(\)](#)(p. 94)
- [amovej\(\)](#)(p. 100)
- [amovel\(\)](#)(p. 103)
- [amovejx\(\)](#)(p. 106)

- [amovec\(\)](#)(p. 109)
- [amovesj\(\)](#)(p. 113)
- [amovesx\(\)](#)(p. 116)
- [amoveb\(\)](#)(p. 119)
- [amove\\_spiral\(\)](#)(p. 124)
- [amove\\_periodic\(\)](#)(p. 128)
- [motion\\_resume\(\)](#)(p. 155)

### 3.5.14 motion\_resume()

#### 기능

일시 정지 된 모션을 재개합니다.

#### 인수

없음

#### 리턴

값	설명
0	성공
음수값	오류

#### 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 amovej([0, 0, 90, 0, 90, 0], 60, 30)
2 wait(1)
3 motion_pause()
4 wait(1)
5 motion_resume()

```

## 관련 명령어

- movej()(p. 57)
- movel()(p. 62)
- movejx()(p. 67)
- movec()(p. 71)
- movesj()(p. 77)
- movesx()(p. 81)
- moveb()(p. 85)
- move\_spiral()(p. 90)
- move\_periodic()(p. 94)
- amovej()(p. 100)
- amovel()(p. 103)
- amovejx()(p. 106)
- amovec()(p. 109)
- amovesj()(p. 113)
- amovesx()(p. 116)
- amoveb()(p. 119)
- amove\_spiral()(p. 124)
- amove\_periodic()(p. 128)
- motion\_pause()(p. 153)

## 3.6 서보 모션

### 3.6.1 servoj()

#### 기능

비동기 방식의 모션 명령어로 모션 시작과 동시에 다음 명령어를 수행합니다. 연속적으로 전달되는 명령 중 가장 최근의 목표 관절 위치를 최대 속도, 가속도 내에서 추종하는 모션입니다.

## 인수

인수명	자료형	기본값	설명
pos	posj	-	posj 또는 joint angle list
	list (float[6])		
vel (v)	float	None	최대 속도(모든 축에 동일) 또는 최대 속도(축별 속도) [deg/s]
	list (float[6])		
acc (a)	float	None	최대 가속도(모든 축에 동일) 또는 최대 가속도(축별 가속도) [deg/s <sup>2</sup> ]
	list (float[6])		
time (t)	float	None	도달 시간 [sec]

### ❶ 알아두기

- 단축 인수를 지원합니다. (v:vel, a:acc, t:time)
- vel이 None인 경우, \_global\_velj가 적용됩니다. (\_global\_velj 초기값은 0이며, set\_velj에 의해 설정 가능)
- acc이 None인 경우, \_global\_accj가 적용됩니다. (\_global\_accj 초기값은 0이며, set\_accj에 의해 설정 가능)
- time이 입력된 경우 최대 속도/가속도 조건에 의해 도달 시간을 지킬 수 없는 경우 자동 조정하고 알림 메시지를 띄웁니다

### ⚠ 주의

- 현재는 속도 슬라이드바의 작동 속도 조절 기능과 연동되지 않습니다.
- 현재는 check\_motion(), change\_operation\_speed() 명령어와 연동되지 않습니다.

## 리턴

값	설명
0	성공

값	설명
음수값	오류

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 set_velj(30)
2 set_accj(60)
3
4 Xt=posj(0, 0, 0, 0, 0, 0)
5 servoj(Xt)
6
7 Xt=posj(0, 0, 0, 0, 0, 0)
8 target = posx(90, 0, 120, -50, 50, -90)
9 del_t = [3, 0.1, 3, -3, 3, -3]
10
11 while 1:
12     for i in range(0,6):
13         Xt[i] = Xt[i] + del_t[i]
14
15         if del_t[i] > 0:
16             if Xt[i] > target[i]:
17                 Xt[i] = target[i]
18         else:
19             if Xt[i] < target[i]:
20                 Xt[i] = target[i]
21
22 servoj(Xt)

```

## 관련 명령어

- [posj\(J1=0, J2=0, J3=0, J4=0, J5=0, J6=0\)\(p. 29\)](#)
- [set\\_velj\(vel\)\(p. 45\)](#)
- [set\\_accj\(acc\)\(p. 47\)](#)
- [mwait\(time=0\)\(p. 132\)](#)

### 3.6.2 servol()

#### 기능

비동기 방식의 모션 명령어로 모션 시작과 동시에 다음 명령어를 수행합니다. 연속적으로 전달되는 명령 중 가장 최근의 목표 위치를 설정한 속도, 가속도 내에서 추종하는 모션입니다.

#### 인수

인수명	자료형	기본값	설명
pos	posx	-	posx 또는 position list
	list (float[6])		
vel (v)	float	None	최대 속도[mm/s] 또는 최대 속도[mm/s], 최대 속도[deg/s]
	list (float[2])		
acc (a)	float	None	최대 가속도[mm/s <sup>2</sup> ] 또는 최대 가속도[mm/s <sup>2</sup> ], 최대 가속도[deg/s <sup>2</sup> ]
	list (float[2])		
time (t)	float	None	도달 시간 [sec]

#### ❶ 알아두기

- 단축 인수를 지원합니다. (v:vel, a:acc, t:time)
- vel이 None인 경우, \_global\_velx가 적용됩니다. (\_global\_velx 초기값은0이며, set\_velx에 의해 설정 가능)
- acc이 None인 경우, \_global\_accx가 적용됩니다. (\_global\_accx 초기값은0이며, set\_accx에 의해 설정 가능)
- time이 입력된 경우 최대 속도/가속도 조건에 의해 도달 시간을 지킬 수 없는 경우 자동 조정하고 알림 메시지를 띄웁니다.

### ⚠ 주의

- 현재는 속도 슬라이드바의 작동 속도 조절 기능과 연동되지 않습니다.
- 현재는 singularity 처리 옵션 중 DR\_VAR\_VEL 옵션과 연동되지 않습니다. DR\_VAR\_VEL 옵션으로 설정 시 자동으로 DR\_AVOID 옵션으로 변경하고 알림 메시지를 띄웁니다.
- 현재는 힘/강성 제어 명령어와 연동되지 않습니다.
- 현재는 check\_motion(), change\_operation\_speed() 명령어와 연동되지 않습니다.

### 리턴

값	설명
0	성공
음수값	오류

### 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

### 예제

```

1 set_velj(30)
2 set_accj(60)
3 set_velx(50)
4 set_accx(100)
5

```

```

6 movej(posj(0,0,90,0,90,0))
7
8 Xt=posx(368, 34.5, 442.5, 50.26, -180, 50.26)
9 servol(Xt,v=[100, 100], a=[200,300])
10
11 Xt=posx(368, 34.5, 442.5, 50.26, -180, 50.26)
12 target =posx(368, 34.5, 200, 50.26, -180, 50.26)
13 del_t = [0, 0, -3, 0, 3, 0]
14
15 while 1:
16     for i in range(0,6):
17         Xt[i] = Xt[i] + del_t[i]
18
19     if del_t[i] > 0:
20         if Xt[i] > target[i]:
21             Xt[i] = target[i]
22         else:
23             if Xt[i] < target[i]:
24                 Xt[i] = target[i]
25 servol(Xt,v=[100, 100], a=[200,300])

```

## 관련 명령어

- posx(x=0, y=0, z=0, A=0, B=0, C=0, ori\_type=None, sol=None, turn=None)(p. 30)
- set\_velx(vel1, vel2)(p. 48)
- set\_accx(acc1, acc2)(p. 50)
- mwait(time=0)(p. 132)

### 3.6.3 speedj()

#### 기능

비동기 방식의 모션 명령어로 모션 시작과 동시에 다음 명령어를 수행합니다. 연속적으로 전달되는 명령 중 가장 최근의 목표 관절 속도를 설정한 최대 속도, 가속도 내에서 추종하는 모션입니다.

#### 인수

인수명	자료형	기본값	설명
vel	list (float[6])	-	목표 관절 속도 [deg/s]
acc (a)	float	None	최대 관절 가속도(모든 축에 동일) 또는
	list (float[6])		최대 관절 가속도(축별 가속도) [deg/s <sup>2</sup> ]
time (t)	float	None	도달 시간 [sec]

### 알아두기

- 단축 인수를 지원합니다. (a:acc, t:time)
- acc이 None인 경우, \_global\_accj가 적용됩니다. (\_global\_accj 초기값은 0이며, set\_accj에 의해 설정 가능)
- time이 입력된 경우, 최대 가속도 조건에 의해 도달 시간을 지킬 수 없는 경우 자동 조정하고 알림 메시지를 띄웁니다.
- 속도가 있는 상태에서 정상적으로 정지하고 싶은 경우에는 vel을 [0,0,0,0,0]으로 입력하거나 stop 명령어를 활용합니다.
- 안전을 위하여 이동 중 1 [sec] 동안 새로운 speedj 명령이 전달되지 않으면, 에러 메시지를 띄우고 정지합니다.

### 주의

- 현재는 속도 슬라이드바의 작동 속도 조절 기능과 연동되지 않습니다.
- 현재는 check\_motion(), change\_operation\_speed() 명령어와 연동되지 않습니다.

## 리턴

값	설명
0	성공
음수값	오류

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시

예외	설명
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1      movej(posj(0,0,90,0,90,0), v=30, a=60)
2
3      v1 = 30
4      go_plus = True
5      while True:
6          q = get_desired_posj()
7          if go_plus:
8              speedj([v1, 5, 5, 5, 5, 5], a=60)
9              if q[0] > 90:
10                 go_plus = False
11             else:
12                 speedj([-v1, -5, -5, -5, -5, -5], a=60)
13                 if q[0] < -90:
14                     go_plus = True

```

## 관련 명령어

- posj(J1=0, J2=0, J3=0, J4=0, J5=0, J6=0)(p. 29)
- set\_velj(vel)(p. 45)
- set\_accj(acc)(p. 47)
- mwait(time=0)(p. 132)
- stop(st\_mode)(p. 138)

## 3.6.4 speedl()

### 기능

비동기 방식의 모션 명령어로 모션 시작과 동시에 다음 명령어를 수행합니다. 연속적으로 전달되는 명령 중 가장 최근의 목표 속도를 설정한 속도, 가속도 내에서 추종하는 모션입니다.

### 인수

인수명	자료형	기본값	설명
vel	list (float[6])	-	목표 속도[mm/s, deg/s]

인수명	자료형	기본값	설명
acc (a)	float	None	최대 가속도[mm/s <sup>2</sup> ] 또는
	list (float[2])		최대 가속도[mm/s <sup>2</sup> ], 최대 가속도[deg/s <sup>2</sup> ]
time (t)	float	None	도달 시간 [sec]

### ❶ 알아두기

- 단축 인수를 지원합니다. (a:acc, t:time)
- acc이 None인 경우, \_global\_accx가 적용됩니다. (\_global\_accx 초기값은 0이며, set\_accx() 의해 설정 가능)
- time이 입력된 경우 최대 가속도 조건에 의해 도달 시간을 지킬 수 없는 경우 자동 조정하고 알림 메시지를 띄웁니다.
- 속도가 있는 상태에서 정상적으로 정지하고 싶은 경우에는 vel을 [0,0,0,0,0,0]으로 입력하거나 stop 명령어를 활용합니다.
- 안전을 위하여 이동 중 1 [sec] 동안 새로운 speedl 명령이 전달되지 않으면, 에러를 발생시키고 정지 합니다.

### ⚠ 주의

- 현재는 속도 슬라이드바의 작동 속도 조절 기능과 연동되지 않습니다.
- 현재는 singularity 처리 옵션 중 DR\_VAR\_VEL 옵션과 연동되지 않습니다. DR\_VAR\_VEL 옵션으로 설정 시 자동으로 DR\_AVOID로 변경하고 알림 메시지를 띄웁니다.
- 현재는 힘/강성 제어 명령어와 연동되지 않습니다.
- 현재는 check\_motion(), change\_operation\_speed() 명령어와 연동되지 않습니다.

### 리턴

값	설명
0	성공
음수값	오류

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1      movej(posj(0,0,90,0,90,0), v=30, a=60)
2      x = get_desired_posx()
3
4      v_y = 250
5      v = 5
6      at_max = 500
7      ar_max = 60
8      go_plus = True
9      while True:
10         xd = get_desired_posx()
11         if go_plus:
12             speedl([v, v_y, v, 30, 0, 30], [at_max, ar_max])
13             if xd[1] > x[1] + 200:
14                 go_plus = False
15         else:
16             speedl([-v, -v_y, -v, -30, -0, -30], [at_max, ar_max])
17             if xd[1] < x[1] - 200:
18                 go_plus = True

```

## 관련 명령어

- posx(x=0, y=0, z=0, A=0, B=0, C=0, ori\_type=None, sol=None, turn=None)(p. 30)
- (3.2.2-ko\_KR) set\_velx(vel)<sup>22</sup>
- (3.2.2-ko\_KR) set\_accx(acc)<sup>23</sup>

<sup>22</sup> [https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/202473924/3.2.2-ko\\_KR+set\\_velx+vel](https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/202473924/3.2.2-ko_KR+set_velx+vel)

<sup>23</sup> [https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/202179059/3.2.2-ko\\_KR+set\\_accx+acc](https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/202179059/3.2.2-ko_KR+set_accx+acc)

- [mwait\(time=0\)\(p. 132\)](#)
- [stop\(st\\_mode\)\(p. 138\)](#)

## 4 제어 보조 명령어

### 4.1 로봇 현재값

#### 4.1.1 get\_current\_posj()

##### 기능

현재 관절 각도를 리턴합니다.

##### 리턴

값	설명
posj	관절각

##### 예외

예외	설명
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시

##### 예제

```
1 q1 = get_current_posj()
```

##### 관련 명령어

- [get\\_desired\\_posj\(\)](#)(p. 176)

#### 4.1.2 get\_current\_velj()

##### 기능

현재 관절 속도를 리턴합니다.

## 리턴

값	설명
float[6]	Joint speed

## 예외

예외	설명
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시

## 예제

```
1 velj1 = get_current_velj()
```

## 관련 명령어

- [get\\_desired\\_velj\(\)](#)(p. 177)

### 4.1.3 get\_current\_posx(ref, ori\_type)

## 기능

현재 태스크 좌표계의 자세와 solution space를 리턴합니다. 이때 자세는 ref coordinate를 기준으로 합니다.

## 인수

인수명	자료형	기본값	설명
ref	Int	DR_BASE	<p>reference coordinate</p> <ul style="list-style-type: none"> <li>• DR_BASE : base coordinate</li> <li>• DR_WORLD : world coordinate</li> <li>• user coordinate: 사용자 정의</li> </ul>

인수명	자료형	기본값	설명
ori_type	int	DR_ELR_ZYZ	<p>orientation type</p> <ul style="list-style-type: none"> <li>• DR_ELR_ZYZ: Euler Angles(z-y'-z", in degrees)</li> <li>• DR_ELR_ZYX: Euler Angles(z-y'-x", in degrees)</li> <li>• DR_ELR_XYZ: Euler Angles(x-y'-z", in degrees)</li> <li>• DR_FIX_XYZ: Fixed Angles(x-y-z, in degrees)</li> <li>• DR_ROTVEC: 3D rotation vector (angle/axis representation)</li> <li>• DR_QUAT: unit quaternion(x, y, z, w)</li> </ul>

### ➊ 알아두기

ref생략시 DR\_BASE로 적용됩니다.

### 리턴

값	설명
Posx	Task space point
Int	Solution space (0 ~ 7)

### Robot configuration vs. solution space

Solution space	Binary	Shoulder	Elbow	Wrist
0	000	Lefty	Below	No Flip
1	001	Lefty	Below	Flip
2	010	Lefty	Above	No Flip
3	011	Lefty	Above	Flip
4	100	Righty	Below	No Flip
5	101	Righty	Below	Flip

Solution space	Binary	Shoulder	Elbow	Wrist
6	110	Righty	Above	No Flip
7	111	Righty	Above	Flip

## 예외

예외	설명
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시

## 예제

```

1 x1, sol = get_current_posx() #x1 w.r.t. DR_BASE
2
3 x1_wld, sol = get_current_posx(ref=DR_WORLD) #x1 w.r.t. DR_WORLD
4
5 DR_USR1=set_user_cart_coord(x1, x2, x3, pos)
6 set_ref_coord(DR_USR1)
7
8 x1, sol = get_current_posx(DR_USR1) #x1 w.r.t. DR_USR1
9
10 x2, sol = get_current_posx(DR_USR1, DR_FIX_XYZ) #x2 w.r.t. DR_USR1
    (orientation type: fixed xyz)

```

## 관련 명령어

- [get\\_desired\\_posx\(ref, ori\\_type\)\(p. 178\)](#)

### 4.1.4 get\_current\_tool\_flange\_posx(ref, ori\_type)

#### 기능

입력된 기준좌표계(ref)에 해당하는 현재 툴 플랜지 포즈를 리턴합니다. 즉, tcp=(0,0,0,0,0,0)인 위치로 리턴하는 것을 의미합니다.

## 인수

인수명	자료형	기본값	설명
ref	Int	DR_BASE	<p>reference coordinate</p> <ul style="list-style-type: none"> <li>• DR_BASE : base coordinate</li> <li>• DR_WORLD : world coordinate</li> </ul>
ori_type	int	DR_ELR_ZYZ	<p>orientation type</p> <ul style="list-style-type: none"> <li>• DR_ELR_ZYZ: Euler Angles(z-y'-z'', in degrees)</li> <li>• DR_ELR_ZYX: Euler Angles(z-y'-x'', in degrees)</li> <li>• DR_ELR_XYZ: Euler Angles(x-y'-z'', in degrees)</li> <li>• DR_FIX_XYZ: Fixed Angles(x-y-z, in degrees)</li> <li>• DR_ROTVEC: 3D rotation vector (angle/axis representation)</li> <li>• DR_QUAT: unit quaternion(x, y, z, w)</li> </ul>

## 리턴

값	설명
posx	Tool flange의 pose

## 예외

예외	설명
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시

## 예제

```

1 x1 = get_current_tool_flange_posx() #x1 : BASE 좌표계(기본값)에서의 flange포즈
2 x2 = get_current_tool_flange_posx(DR_BASE) #x2: BASE 좌표계에서의 flange포즈
3 x3 = get_current_tool_flange_posx(DR_WORLD) #x3: WORLD좌표계에서의 flange포즈
4 x4 = get_current_tool_flange_posx(ori_type=DR_FIX_XYZ) #x4: BASE 좌표계(기본값)에서의 flange포즈 (orientation type: fixed xyz)

```

## 4.1.5 get\_current\_velx(ref)

### 기능

입력된 기준좌표계(ref)에 해당하는 현재 툴 속도를 리턴합니다.

### 인수

인수명	자료형	기본값	설명
ref	Int	DR_BASE	reference coordinate <ul style="list-style-type: none"> <li>• DR_BASE : base coordinate</li> <li>• DR_WORLD : world coordinate</li> </ul>

### 리턴

값	설명
float[6]	Tool velocity

### 예외

예외	설명
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시

### 예제

```

1 velx1 = get_current_velx() # velx1 : BASE좌표계(기본값)의 속도
2 velx2 = get_current_velx(DR_BASE) # velx2 (=velx1) : BASE좌표계의 속도
3 velx3 = get_current_velx(DR_WORLD) #velx3 : WORLD좌표계의 속도

```

### 관련 명령어

- [get\\_desired\\_velx\(ref\)\(p. 179\)](#)

## 4.1.6 get\_current\_rotm(ref)

### 기능

입력된 기준좌표계(ref)에 해당하는 현재 툴의 회전행렬을 리턴합니다.

### 인수

인수명	자료형	기본값	설명
ref	Int	DR_BASE	reference coordinate <ul style="list-style-type: none"> <li>• DR_BASE : base coordinate</li> <li>• DR_WORLD : world coordinate</li> </ul>

### 리턴

값	설명
float[3][3]	Rotation matrix

### 예외

예외	설명
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시

### 예제

```

1   rotm1 = get_current_rotm(DR_WORLD) #rotm1 : WORLD좌표계 기준 회전행렬(3x3)
2   # 결과값은 3X3 matrix로 저장됩니다.

```

$$\text{rotm1} = \begin{bmatrix} \text{rotm1}[0][0] & \text{rotm1}[0][1] & \text{rotm1}[0][2] \\ \text{rotm1}[1][0] & \text{rotm1}[1][1] & \text{rotm1}[1][2] \\ \text{rotm1}[2][0] & \text{rotm1}[2][1] & \text{rotm1}[2][2] \end{bmatrix}$$

## 4.1.7 get\_joint\_torque()

### 기능

현재 조인트의 센서 토크 값을 리턴합니다.

### 리턴

값	설명
float[6]	JTS 토크값

### 예외

예외	설명
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시

### 예제

```
1 j_trq1 = get_joint_torque()
```

### 관련 명령어

- [get\\_external\\_torque\(\)](#)(p. 174)
- [get\\_tool\\_force\(ref\)](#)(p. 175)

## 4.1.8 get\_external\_torque()

### 기능

현재 각 관절에서 외력에 의해 발생하는 토크 값을 리턴합니다.

### 리턴

값	설명
float[6]	외력에 의해 발생하는 토크값

## 예외

예외	설명
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시

## 예제

```
1 trq_ext=get_external_torque()
```

## 관련 명령어

- [get\\_joint\\_torque\(\)](#)(p. 174)
- [get\\_tool\\_force\(ref\)](#)(p. 175)

## 4.1.9 get\_tool\_force(ref)

### 기능

입력된 기준좌표계(ref)에서의 현재 툴에 작용하는 외력 값을 리턴합니다. 출력 값의 힘(Force), <sup>1)</sup>모멘트(Moment)는 기준좌표계(ref) 기준으로 합니다.

<sup>1)</sup>V2.8 이전 버전에서 모멘트(Moment)는 Tool 좌표계를 기준으로 합니다

### 인수

인수명	자료형	기본값	설명
ref	Int	DR_BASE	reference coordinate <ul style="list-style-type: none"> <li>• DR_BASE : base coordinate</li> <li>• DR_WORLD : world coordinate</li> <li>• DR_TOOL : tool coordinate</li> </ul>

### 리턴

값	설명
float[6]	Tool에 작용하는 외력

## 예외

예외	설명
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시

## 예제

```
1 force_ext = get_tool_force(DR_WORLD) # force_ext: WORLD좌표계 기준 툴의 외력
```

## 관련 명령어

- [get\\_joint\\_torque\(\)](#)(p. 174)
- [get\\_external\\_torque\(\)](#)(p. 174)

## 4.2 로봇 목표값

### 4.2.1 get\_desired\_pos()

#### 기능

현재의 목표(target) 관절각을 리턴합니다. 단, movel, movec, movesx, moveb, move\_spiral, move\_periodic 명령어에서는 사용할 수 없습니다.

#### 리턴

값	설명
posj	관절각

## 예외

예외	설명
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_INVALID)	유효하지 않은 명령어

**예제**

1	<code>jp1 = get_desired_posj()</code>
---	---------------------------------------

**관련 명령어**

- `get_current_posj()`(p. 167)

**4.2.2 get\_desired\_velj()****기능**

현재의 목표(target) 관절 속도를 리턴합니다. 단, movel, movec, movesx, moveb, move\_spiral, move\_periodic 명령어에서는 사용할 수 없습니다.

**리턴**

값	설명
<code>float[6]</code>	목표 관절 속도

**예외**

예외	설명
<code>DR_Error (DR_ERROR_RUNTIME)</code>	C Extension 모듈 에러 발생 시
<code>DR_Error (DR_ERROR_INVALID)</code>	유효하지 않은 명령어

**예제**

1	<code>velj1 = get_desired_velj()</code>
---	---

**관련 명령어**

- `get_current_velj()`(p. 167)

### 4.2.3 get\_desired\_posx(ref, ori\_type)

#### 기능

현재의 툴의 목표(target) 자세를 리턴합니다. 이때 자세는 ref coordinate 기준으로 합니다.

#### 인수

인수명	자료형	기본값	설명
ref	Int	DR_BASE	<p>reference coordinate</p> <ul style="list-style-type: none"> <li>• DR_BASE : base coordinate</li> <li>• DR_WORLD : world coordinate</li> <li>• user coordinate: 사용자 정의</li> </ul>
ori_type	int	DR_ELR_ZYZ	<p>orientation type</p> <ul style="list-style-type: none"> <li>• DR_ELR_ZYZ: Euler Angles(z-y'-z'', in degrees)</li> <li>• DR_ELR_ZYX: Euler Angles(z-y'-x'', in degrees)</li> <li>• DR_ELR_XYZ: Euler Angles(x-y'-z'', in degrees)</li> <li>• DR_FIX_XYZ: Fixed Angles(x-y-z, in degrees)</li> <li>• DR_ROTVEC: 3D rotation vector (angle/axis representation)</li> <li>• DR_QUAT: unit quaternion(x, y, z, w)</li> </ul>

#### ● 알아두기

ref생략시 DR\_BASE로 적용됩니다.

#### 리턴

값	설명
float[6]	Tool velocity

#### 예외

예외	설명
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시

## 예제

```

1 x1 = get_desired_posx() #x1 w.r.t. DR_BASE
2 x2 = posx(100, 0, 0, 0, 0, 0)
3 x3 = posx(0, 0, 20, 20, 20, 20)
4 pos = x3
5 DR_USR1=set_user_cart_coord(x1, x2, x3, pos)
6 set_ref_coord(DR_USR1)
7
8 xa = get_desired_posx(DR_USR1) #xa w.r.t. DR_USR1
9 xb = get_desired_posx(DR_WORLD, DR_ELR_ZYX) #xb w.r.t. DR_WORLD,
orientation type: euler zyx

```

## 관련 명령어

- [get\\_current\\_posx\(ref, ori\\_type\)\(p. 168\)](#)

### 4.2.4 get\_desired\_velx(ref)

#### 기능

입력된 기준좌표계(ref)에 해당하는 현재 툴의 목표(target) 속도를 리턴합니다. 단, movej, movejx, movesj 명령어에서는 사용할 수 없습니다.

#### 인수

인수명	자료형	기본값	설명
ref	Int	DR_BASE	reference coordinate <ul style="list-style-type: none"> <li>• DR_BASE : base coordinate</li> <li>• DR_WORLD : world coordinate</li> </ul>

#### 리턴

값	설명
float[6]	Tool velocity

## 예외

예외	설명
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_INVALID)	유효하지 않은 명령어

## 예제

```

1 vel_x1 = get_desired_velx() #vel_x1 : BASE좌표계(기본값)에서의 툴의 목표속도
2 vel_x2 = get_desired_velx(DR_BASE) #vel_x2 : BASE좌표계에서의 툴의 목표속도
3 vel_x3 = get_desired_velx(DR_WORLD) #vel_x3 : WORLD좌표계에서의 툴의 목표속도

```

## 관련 명령어

- [get\\_current\\_velx\(ref\)\(p. 172\)](#)

## 4.3 제어 상태값

### 4.3.1 get\_control\_mode()

#### 기능

현재 제어 모드를 리턴합니다.

#### 리턴

값	설명
int	제어모드 3 : Position control mode 4 : Torque control mode

## 예외

예외	설명
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시

**예제**

1	<code>mode = get_control_mode()</code>
---	--

**4.3.2 get\_control\_space()****기능**

현재 제어 공간을 리턴합니다.

**리턴**

값	설명
int	제어모드 1 : Joint space control 2 : Task space control

**예외**

예외	설명
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시

**예제**

1	<code>x1 = get_control_space()</code>
---	---------------------------------------

**4.3.3 get\_current\_solution\_space()****기능**

현재 solution space 값을 리턴합니다.

## 리턴

값	설명
int	Solution space (0 ~ 7)

## 예외

예외	설명
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시

## 예제

```
1 sol = get_current_solution_space()
```

## 관련 명령어

- [get\\_solution\\_space\(pos\)\(p. 182\)](#)

## 4.3.4 get\_solution\_space(pos)

### 기능

입력된 pos(posj)에 대한 Solution space 값을 구합니다.

## 인수

인수명	자료형	기본값	설명
pos	posj	-	posj 또는 position list
	list (float[6])		

## 리턴

값	설명
0 ~ 7	Solution space

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 q1 = posj(0, 0, 0, 0, 0, 0)
2 sol1 = get_solution_space(q1)
3 sol2 = get_solution_space([10, 20, 30, 40, 50, 60])

```

## 관련 명령어

- `get_current_solution_space()`(p. 181)

## 4.3.5 `get_orientation_error(xd, xc, axis)`

### 기능

`axis`에 대한 임의의 pose `xd`와 `xc` 사이의 Orientation error 값을 리턴합니다.

### 인수

인수명	자료형	기본값	설명
<code>xd</code>	<code>posx</code>	-	<code>posx</code> 또는 <code>position list</code>
	<code>list (float[6])</code>		
<code>xc</code>	<code>posx</code>	-	<code>posx</code> 또는 <code>position list</code>
	<code>list (float[6])</code>		

인수명	자료형	기본값	설명
axis	int	-	axis <ul style="list-style-type: none"> <li>• DR_AXIS_X: x축</li> <li>• DR_AXIS_Y: y축</li> <li>• DR_AXIS_Z: z축</li> </ul>

## 리턴

값	설명
float	Orientation error 값

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1  xd = posx(0, 0, 0, 0, 0, 0)
2  xc = posx(10, 20, 30, 40, 50, 60)
3  diff = get_orientation_error(xd, xc, DR_AXIS_X)

```

## 관련 명령어

- [get\\_current\\_rotm\(ref\)\(p. 173\)](#)

## 5 기타 설정 명령어

### 5.1 툴/작업물 설정

#### 5.1.1 get\_workpiece\_weight()

##### 기능

작업물의 무게를 측정하여 리턴합니다.

**⚠ 주의**

해당 명령어는 Non-FTS A 모델에서는 사용이 불가합니다.

##### 리턴

값	설명
0 이상의 값	측정 무게 값
음수값	오류

##### 예외

예외	설명
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

##### 예제

1	weight = get_workpiece_weight()
---	---------------------------------

##### 관련 명령어

- reset\_workpiece\_weight()(p. 186)

## 5.1.2 reset\_workpiece\_weight()

### 기능

소재의 무게를 측정하기 전 알고리즘의 초기화를 위해 소재의 무게정보를 초기화합니다.

### 리턴

값	설명
0	성공
음수값	오류

### 예외

예외	설명
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

### 예제

1	reset_workpiece_weight()
---	--------------------------

### 관련 명령어

- [get\\_workpiece\\_weight\(\)](#)(p. 185)

## 5.1.3 set\_workpiece\_weight(weight=0.0, cog=[0.0,0.0,0.0], cog\_ref=DR\_CUR\_TCP, add\_up=DR\_REPLACE, start\_time=None, transition\_time=None)

### 기능

로봇 끝단의 틀 무게/무게 중심에 추가하여 작업물의 무게/무게 중심, 기타 정보를 설정합니다. 전체 페이로드의 무게 및 무게 중심은 설정된 틀 무게/무게 중심과 작업물의 무게/무게 중심을 종합하여 반영됩니다. 작업물의 종류가 다양하거나 동적으로 무게가 변경되어야 하는 Application에서 쓸 수 있습니다.

### 주의

- 작업률 무게 변경은 Auto Mode 중 Collision Detection과 TCP SLF Violation 검사가 모두 무효화된 경우에만 허용됩니다.
- 현재 버전에서, Collision Detection은 Collision Sensitivity를 0으로 Override 한 경우, TCP SLF Violation은 TCP SLF Limit을 최대로 Override 한 경우를 기능이 무효화된 상태로 간주합니다. 이 무효화는 Collision Sensitivity Reduction Zone, Custom Zone 등을 이용해서 설정할 수 있습니다.
- 그 외의 경우에 작업률 무게가 0으로 설정되어 있지 않으면 ss1 보호 정지를 발생시킵니다.
- 에러 발생으로 로봇이 정지하여 매뉴얼로 복구하여야 하는 경우 복구 모드에서 로봇을 원하는 위치에 놓고, Auto Mode에서 해당 Zone들이 활성화된 상태에서 Servo On, I/O 조작 등을 통해 작업률을 언로딩할 수 있습니다.
- 설정된 툴 무게를 변경하는 경우에는 작업률 무게가 0으로 초기화 됩니다.

## 인수

인수명	자료형	기본값	설명
weight	int	0	무게 [kgf]
cog	list(float[3])	[0, 0, 0]	무게 중심 위치(x, y, z) [mm]
cog_ref	int	DR_CUR_TCP	무게 중심 위치 기준 좌표계, DR_CUR_TCP : TCP 기준, DR_FLANGE : FLANGE 기준,
add_up	int	DR_REPLACE	DR_REPLACE(0): 작업률 교체 DR_ADD(1): 작업률 추가 DR_REMOVE(2): 작업률 제거
start_time	float	None	전달 시간 이후 작업률 무게 변경 시작[sec]
transition_time	float	None	전달 시간 동안 작업률 무게 점진적 변경[sec]

### 알아두기

- 작업률의 무게는 모델별 최대 페이로드 무게(margin 10%)를 초과할 수 없습니다. 전체 페이로드의 무게도 마찬가지입니다.
- 작업률의 무게 중심(x, y, z) 각 축의 길이는 1000mm를 초과할 수 없습니다. 전체 페이로드의 무게 중심 각 축의 길이도 마찬가지입니다.
- start\_time을 통해 해당 설정 시간 이후에 작업률 무게 변경이 가능합니다.

- transition\_time을 통해 해당 설정 시간 동안 작업물 무게를 점진적으로 변경이 가능합니다.
- set\_tool, set\_workpiece\_weight 함수를끼리 연달아 사용하는 경우에는 transition\_time 만큼 wait(transition\_time) 함수를 사이에 넣어서 사용해야 합니다. 그렇지 않으면 무게 변경에 오류가 생길 수 있습니다.

## 리턴

값	설명
0	성공
음수값	오류

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 # Replace 1 kgf workpiece weight at the origin of the current TCP frame
2 set_workpiece_weight(1)
3 # Add 0.5 kgf workpiece weight at the origin of the current TCP frame. In
4 # total, 1.5kgf
5 set_workpiece_weight(0.5, add_up=DR_ADD)
6 # Remove 0.1 kgf workpiece weight at [0, 0, 10] in the current TCP frame.
7 # In total, 1.4kgf
8 set_workpiece_weight(0.1, [0, 0, 10], add_up=DR_REMOVE)
9 # Remove 0.1 kgf workpiece weight at [0, 0, 10] in the flange frame. In
10 # total, 1.3kgf
11 set_workpiece_weight(0.1, [0, 0, 10], DR_FLANGE, DR_REMOVE)
# Reset. And the weight transition is being occurred after 0.1 sec for 0.2
sec
set_workpiece_weight(0, start_time=0.1, transition_time=0.2)

```

## 관련 명령어

- `set_tool(name, start_time, transition_time)`(p. 190)

### 5.1.4 `set_tool_shape(name)`

#### 기능

티치펜던트에 등록된 툴 형상 정보 중에서 입력된 name의 tool 형상을 활성화합니다.

#### 인수

인수명	자료형	기본값	설명
name	string	-	티치펜던트에 등록된 툴 shape 이름

#### 리턴

값	설명
0	성공
음수값	오류

#### 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

#### 예제

```
1 set_tool_shape("tool_shape1") #TP에서 등록된 "tool_shape1"의 정보를 활성화한다.
```

## 관련 명령어

- [set\\_tcp\(name\)\(p. 52\)](#)

### 5.1.5 set\_tool(name, start\_time, transition\_time)

#### 기능

티치펜던트 > 워크셀 매니저(Workcell Manager) > 로봇에 등록된 툴 무게(Tool Weight) 워크셀 아이템 중에서 입력된 name의 툴 무게(Tool Weight) 워크셀 아이템을 활성화합니다.

start\_time[sec]을 통해 해당 설정 시간 이후에 툴 무게 변경이 가능합니다. transition\_time[sec]을 통해 해당 설정 시간 동안 툴 무게를 점진적으로 변경이 가능합니다.

#### 인수

인수명	자료형	기본값	설명
name	string	-	워크셀매니저에 등록된 툴 무게(Tool Weight)의 이름
start_time	float	None	전달 시간 이후 툴 무게 변경 시작
transition_time	float	None	전달 시간 동안 툴 무게 점진적 변경

#### ❶ 알아두기

- set\_tool, set\_workpiece\_weight 함수를끼리 연달아 사용하는 경우에는 transition\_time 만큼 wait(transition\_time) 함수를 사이에 넣어서 사용해야 합니다. 그렇지 않으면 무게 변경에 오류가 생길 수 있습니다.

#### 리턴

값	설명
0	성공
음수값	오류

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 set_tool ("tool1",start_time=1,transition_time=2)
2 # 1초 후, 2초 동안 TP에서 등록된 "tool1"의 정보를 활성화 한다. 정보를 반영한다.

```

## 관련 명령어

- [set\\_tcp\(name\)\(p. 52\)](#)

## 5.2 제어 모드 설정

### 5.2.1 set\_singularity\_handling(mode)

#### 기능

task motion에서 특이점의 영향으로 path deviation이 발생할 경우 대응 정책을 사용자가 선택할 수 있도록 합니다. mode의 설정은 아래와 같은 설정이 가능 합니다.

- 자동회피 모드(Default) : DR\_AVOID
- 경로 우선 : DR\_TASK\_STOP
- 속도 가변 : DR\_VAR\_VEL

기본 설정은 자동회피 모드이며, 이 설정의 경우 특이점으로 인한 불안정성을 감소시키지만 path tracking 정확도가 감소합니다. 경로 우선 설정의 경우 singularity의 영향으로 불안정성이 발생할 가능성이 있는 경우, 감속 후 warning 메시지를 출력하고 해당 Task를 종료합니다. 속도 가변 설정의 경우 특이점으로 인한 불안정을 감소시키면서 path tracking 정확도를 높입니다. 하지만 특이점 구간에서 TCP 속도 변경이 발생합니다

## 인수

인수명	자료형	기본값	설명
mode	int	DR_AVOID	DR_AVOID : 자동 회피 모드 DR_TASK_STOP : 감속/ Warning/ Task 종료 DR_VAR_VEL : 속도 가변

## 리턴

값	설명
0	성공
음수값	오류

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 P1 = posx(400,500,800,0,180,0)
2 P2 = posx(400,500,500,0,180,0)
3 P3 = posx(400,500,200,0,180,0)
4 set_singularity_handling(DR_AVOID) # 특이점 자동회피 모드
5 movel(P1, vel=10, acc=20)
6 set_velx(30)
7 set_accx(60)
8 set_singularity_handling(DR_TASK_STOP) # Task 모션 경로 우선
9 movel(P2)
10 set_singularity_handling(DR_VAR_VEL) # 특이점 속도 가변 모드
11 movel(P3)

```

## 관련 명령어

- [move\(\)](#)(p. 62)
- [movec\(\)](#)(p. 71)
- [movesx\(\)](#)(p. 81)
- [moveb\(\)](#)(p. 85)
- [move\\_spiral\(\)](#)(p. 90)
- [amovel\(\)](#)(p. 103)
- [amovec\(\)](#)(p. 109)
- [amovesx\(\)](#)(p. 116)
- [amoveb\(\)](#)(p. 119)
- [amove\\_spiral\(\)](#)(p. 124)

## 5.2.2 set\_singular\_handling\_force(mode)

### 기능

기본 설정은 특이점 영역 내에서 컴플라이언스 및 힘 제어 사용 시 에러 처리를 통해 프로그램이 종료됩니다. Mode 설정을 통해 특이점 영역 내에서 에러 처리를 무시할 수 있습니다.

- 에러 처리 : DR\_SINGULARITY\_ERROR
- 에러 처리 무시 : DR\_SINGULARITY\_IGNORE

### ⚠ 주의

특이점 영역 내에서 컴플라이언스 및 힘 제어 사용을 권장하지 않습니다. 특정 방향으로 힘 추정이 정확하지 않을 수 있습니다.

### 인수

인수명	자료형	기본값	설명
mode	int	DR_SINGULARITY_ERROR	DR_SINGULARITY_ERROR : 에러 처리 DR_SINGULARITY_IGNORE : 에러 처리 무시

### 리턴

값	설명
0	성공

값	설명
음수값	오류

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 set_ref_coord(DR_BASE)
2 P0 = posj(0, 0, 90, 0, 90, 0)
3 movej(P0, vel=30, acc=60)
4
5 # 특이점 진입 시 에러 처리 무시
6 set_singular_handling_force(DR_SINGULARITY_IGNORE)
7
8 task_compliance_ctrl()
9 set_stiffnessx([500, 500, 500, 100, 100, 100], time=0.5)
10 fd = [0, 0, 30, 0, 0, 0]
11 fctrl_dir= [0, 0, 1, 0, 0, 0]
12 set_desired_force(fd, dir=fctrl_dir, mod=DR_FC_MOD_REL)
13 release_compliance_ctrl()

```

## 관련 명령어

- task\_compliance\_ctrl(stx, time)(p. 200)
- set\_stiffnessx(stx, time)(p. 201)
- set\_desired\_force(fd, dir, time, mod)(p. 203)
- release\_compliance\_ctrl()(p. 199)

### 5.2.3 set\_palletizing\_mode(mode)

#### 기능

팔레이징 응용 모션에서 wrist 특이점 근방에서 설정한 위치와 속도를 정확히 지킬 수 있는 모드입니다. 모션 지령 중 B방향 성분은 0deg 혹은 180deg로 설정한 상태에서 사용할 때 wrist 특이점 근방에서 안정적으로 사용이 가능합니다.

- 모드 비설정 : DR\_OFF
- 모드 설정 : DR\_ON

#### **⚠ 주의**

- 툴 방향 설정 시 B방향 성분은 0deg 혹은 180deg로 설정합니다. 이때 이 함수 사용이 가능하며 만족하지 않고 함수 사용시 에러가 발생합니다.
- 이 함수 사용 시 속도 변경이 일어나지 않지만 허용 가능한 최대 조인트 속도를 넘을 시 자동으로 속도가 감속됩니다.
- H시리즈 모델의 경우 함수 내에서 힘 제어 사용 시 Rx, Ry 방향 모멘트 제어는 제한이 됩니다. Rx, Ry 방향 외력 모멘트 값은 0 입니다

#### 인수

인수명	자료형	기본값	설명
mode	int	DR_ON	DR_OFF(0) : 모드 비설정 DR_ON(1) : 모드 설정

#### 리턴

값	설명
0	성공
음수값	오류

#### 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시

예외	설명
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 set_singularity_handling(DR_VAR_VEL)
2
3 movej(posj(0,0,90,0,90,0),vel=30,acc=60)
4
5 set_palletizing_mode(DR_ON)
6 movel(posx(559,34.5,-400,45,180,45),vel=500,acc=1000)
7 set_palletizing_mode(DR_OFF)

```

## 관련 명령어

- [set\\_singularity\\_handling\(mode\)\(p. 191\)](#)

### 5.2.4 set\_motion\_end(mode)

#### 기능

모션 완료후 로봇의 정지 상태를 확인하는 기능의 동작 여부를 설정하는 함수입니다. 기능을 정지하면 연속한 모션의 모션 사이 정지하는 구간에서 소요하는 시간이 감소하여 전체 작업시간을 줄이기 위한 목적으로 활용이 가능합니다. 툴의 무게가 무겁고, 높은 가속도로 구동하는 모션 명령어에서 정확한 정지 위치가 필요한 경우에는 기능을 활성화하는 것을 권장합니다.

#### ⚠ 주의

- 모션블랜딩을 통해 정지없이 연속적으로 동작하는 모션 사이에서는 동작 여부를 변경할 수 없습니다.
- 정지상태가 필요없는 연속 모션의 경우에는 모션블랜딩을 사용하는 것이 작업시간을 감소하는 것에 보다 효과적입니다.
- 프로그램 종료후에는 다시 기본값으로 초기화 됩니다.

## 인수

인수명	자료형	기본값	설명
mode	int	DR_CHECK_ON	DR_CHECK_OFF(0) : 모드 비설정 DR_CHECK_ON(1) : 모드 설정

## 리턴

값	설명
0	성공
음수값	오류

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 set_motion_end(DR_CHECK_OFF)
2
3 movej(posj(0,0,90,0,90,0) ,vel=30,acc=60,mod = DR_MV_MOD_ABS )
4 while 1:
5     movej(posj(0,0,10,0,10,0) ,vel=30,acc=60,,mod = DR_MV_MOD_REL )
6     movej(posj(0,0,-10,0,-10,0) ,vel=30,acc=60,,mod = DR_MV_MOD_REL )

```

## 관련 명령어

- [movej\(\)](#)(p. 62)
- [movec\(\)](#)(p. 71)
- [movesx\(\)](#)(p. 81)

- [moveb\(\)](#)(p. 85)
- [move\\_spiral\(\)](#)(p. 90)
- [amovel\(\)](#)(p. 103)
- [amovec\(\)](#)(p. 109)
- [amovesx\(\)](#)(p. 116)
- [amoveb\(\)](#)(p. 119)
- [amove\\_spiral\(\)](#)(p. 124)

## 6 힘/순응 제어 및 기타 사용자 편의 기능

### 6.1 힘/순응 제어

#### 6.1.1 release\_compliance\_ctrl()

##### 기능

Compliance control을 종료하고 현재 위치에서 위치 제어를 시작합니다.

##### 리턴

값	설명
0	성공
음수값	오류

##### 예외

예외	설명
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

##### 예제

```

1 P0 = posj(0,0,90,0,90,0)
2 movej(P0)
3 task_compliance_ctrl()
4 set_stiffnessx([100, 100, 300, 100, 100, 100])
5 release_compliance_ctrl()

```

##### 관련 명령어

- task\_compliance\_ctrl(stx, time)(p. 200)
- set\_stiffnessx(stx, time)(p. 201)

## 6.1.2 task\_compliance\_ctrl(stx, time)

### 기능

기준에 설정한 기준 좌표계를 기준으로 태스크 Compliance control을 시작합니다.

**⚠ 주의**

Non-FTS A 모델에서는 stx 파라미터의 자료형이 float[3]으로 변경됩니다(회전 강성 입력 불가)

**⚠ 주의**

로봇 없이 시뮬레이션 환경에서 해당 명령어를 사용하는 경우, 정상적으로 동작하지 않을 수 있습니다.

### 인수

인수명	자료형	기본값	설명
stx	float[6]	[3000, 3000, 3000, 200, 200, 200]	Translational 강성 3개, 회전강성 3개
time	float	0	강성변화 시간 [sec] 범위 0~1.0 * 주어진 시간 동안 linear transition

### 리턴

값	설명
0	성공
음수값	오류

### 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시

예외	설명
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 P0 = posj(0,0,90,0,90,0)
2 movej(P0)
3 task_compliance_ctrl() # default 강성으로 시작
4 set_stiffnessx([500, 500, 500, 100, 100, 100], time=0.5)
5 # 사용자 정의 강성으로 0.5초 동안 전환
6 release_compliance_ctrl()
7
8 task_compliance_ctrl([500, 500, 500, 100, 100, 100])
9 # 사용자 정의 강성으로 시작
10 release_compliance_ctrl()
```

## 관련 명령어

- [set\\_stiffnessx\(stx, time\)](#)(p. 201)
- [release\\_compliance\\_ctrl\(\)](#)(p. 199)

### 6.1.3 set\_stiffnessx(stx, time)

#### 기능

전역으로 설정된 좌표계(set\_ref\_coord() 참조) 기준으로 강성값을 설정합니다. 현재 강성 또는 기본값으로부터 STX로 주어진 time값 동안 linear transition 합니다.

#### ❶ 강성 사용자 범위

- M/H 시리즈 : Translation(0~20000N/m), Rotation(0~1000Nm/rad)
- A 시리즈 : Translation(0~10000N/m), Rotation(0~300Nm/rad)

#### ⚠ 주의

Non-FTS A 모델에서는 stx 파라미터의 자료형이 float[3]으로 변경됩니다(회전 강성 입력 불가)

#### ⚠ 주의

로봇 없이 시뮬레이션 환경에서 해당 명령어를 사용하는 경우, 정상적으로 동작하지 않을 수 있습니다.

## 인수

인수명	자료형	기본값	설명
stx	float[6]	[500, 500, 500, 100, 100, 100]	Translational 강성3개, 회전강성 3개
time	float	0	강성변화 시간 [sec] 범위 0~1.0 * 주어진 시간 동안 linear transition

## 리턴

값	설명
0	성공
음수값	오류

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 set_ref_coord(DR_WORLD) # 전역좌표계를 World로 설정
2 x0 = posx(0, 0, 90, 0, 90, 0)
3 movej(x0)
4 task_compliance_ctrl()
5 stx = [1, 2, 3, 4, 5, 6]
6 set_stiffnessx(stx) # 현재의 전역좌표계(World) 기준 강성 적용
7 release_compliance_ctrl()
```

## 관련 명령어

- [task\\_compliance\\_ctrl\(stx, time\)\(p. 200\)](#)
- [release\\_compliance\\_ctrl\(\)\(p. 199\)](#)

### 6.1.4 set\_desired\_force(fd, dir, time, mod)

#### 기능

전역으로 설정된 좌표계(set\_ref\_coord() 참조) 기준으로 힘 제어 목표값, 힘 제어 방향, 힘 목표값, 외력참조모드를 설정합니다.

#### 주의

Non-FTS A 모델에서는 fd, dir 파라미터의 자료형이 float[3]으로 변경됩니다.(Rotational parameter 입력 불가)

#### 주의

로봇 없이 시뮬레이션 환경에서 해당 명령어를 사용하는 경우, 정상적으로 동작하지 않을 수 있습니다.

#### 인수

인수명	자료형	기본값	설명
Fd	float[6]	[0, 0, 0, 0, 0, 0]	(Translational) 힘 성분 3개, (Rotational) 모멘트 성분 3개
dir	int[6]	[0, 0, 0, 0, 0, 0]	1이면 해당 방향 힘 제어 0이면 해당 방향 compliance 제어
time	float	0	힘을 증가시키는데 소요되는 시간 [sec] 범위 0~1.0
mod	int	DR_FC_MOD_ABS	DR_FC_MOD_ABS(0): 힘제어 시 외력을 센서값 그대로(절대적) 참조 DR_FC_MOD_REL(1): 힘제어 초기의 센서값을 기준으로 상대적인 외력만 참조

## 리턴

값	설명
0	성공
음수값	오류

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

### 알아두기

- release\_force() 명령을 통한 힘제어 종료(강성제어 전환)시 외력값은 센서값을 참조합니다. 따라서 mod=DR\_FC\_MOD\_REL 옵션을 선택한 경우에 참조하는 외력값의 변화가 발생합니다.
- mod를 DR\_FC\_MOD\_REL로 설정하더라도 충돌 감지나 툴 무게 추정에서는 센서값을 참조합니다

### 주의

힘제어 시의 정확도를 확보하기 위해서는 접촉할 대상물에 근접하여 mod=DR\_FC\_MOD\_REL 옵션을 설정하고 힘제어를 시작하며, 또한 힘제어 중 제한된 영역에서 위치/자세를 변화시키는 것을 권장합니다.

## 예제

```

1 # Example # 1
2 # 전역좌표계(Tool) 기준으로 힘제어 수행
3 # Tool-z축방향 zero force제어, Tool-z축 모멘트 제어, 나머지축 Compliance 제어
4 # Relative Force Mode 활성화 (초기외력 기준의 상대적 외력을 참조하여 힘제어)
5 set_ref_coord(DR_TOOL)
6 x0 = posx(0, 0, 90, 0, 90, 0)
7 movej(x0)
8 task_compliance_ctrl(stx=[500, 500, 500, 100, 100, 100])
9 fd = [0, 0, 0, 0, 0, 10]
10 fctrl_dir= [0, 0, 1, 0, 0, 1]
11 set_desired_force(fd, dir=fctrl_dir, mod=DR_FC_MOD_REL)

```

```

12
13 # Example #2
14 # 초기 자세 이동: [J1, J2, J3, J4, J5, J6] = [0, 0, 90, 0, 90, 0]
15 # 1. 힘 제어를 수행하고자 하는 대상 위치로 Approach: 초기 자세(1)에서
16 # (Base 기준) -z 방향으로 100mm 이동
17 # 2. 힘 제어 수행: (Base 기준) -z 방향으로 20N의 힘 인가
18 # 3. 외력 감지 후 힘-강성 제어: 외력을 가하고 있는 방향(Base 기준 z 방향)에서
19 # 20N의 힘이 감지되면, 기존의 힘 제어를 유지하면서 (Base 기준) y 방향으로
20 # 200mm 이동하는 힘-강성 동시 제어 수행
21 # 4. 힘-강성 제어 종료 후 Retract 및 초기 자세 이동: 힘 제어 및 강성 제어 종료
22 # 후 (Base 기준) +z 방향으로 150mm 이동하여 Retract, Retract 완료 후 초기
23 # 자세(1)로 이동
24
25 # 1. 초기 자세 이동
26 q0 = posj(0.0, 0.0, 90.0, 0.0, 90.0, 0.0)
27 set_velj(30.0)
28 set_accj(60.0)
29 movej(q0)
30
31 # 2. 힘 제어 대상 위치 접근
32 set_velx(75.0)
33 set_accx(100.0)
34 delta_approach = [0.0, 0.0, -100.0, 0.0, 0.0, 0.0]
35 movel(delta_approach, mod=DR_MV_MOD_REL)
36
37 # 3. 힘 제어 수행 (Base 좌표계 기준 -z 방향 힘 인가)
38 k_d = [3000.0, 3000.0, 3000.0, 200.0, 200.0, 200.0]
39 task_compliance_ctrl(k_d)
40 force_desired = 20.0
41 f_d = [0.0, 0.0, -force_desired, 0.0, 0.0, 0.0]
42 f_dir = [0, 0, 1, 0, 0, 0]
43 set_desired_force(f_d, f_dir)
44
45 # 4. 외력 감지 후 힘-강성 제어 수행
46 force_check = 20.0
47 force_condition = check_force_condition(DR_AXIS_Z, max=force_check)
48 while (force_condition):
49     force_condition = check_force_condition(DR_AXIS_Z, max=force_check)
50     if force_condition == 0:
51         break
52     delta_motion = [0.0, 200.0, 0.0, 0.0, 0.0, 0.0]
53     movel(delta_motion, mod=DR_MV_MOD_REL)
54
55 # 5. 힘-강성 제어 종료 후 Retract 및 초기 자세 이동
56 release_force()
57 wait(0.5)
58 delta_retract = [0.0, 0.0, 150.0, 0.0, 0.0, 0.0]
59 release_compliance_ctrl()
60 movel(delta_retract, mod=DR_MV_MOD_REL)
61 movej(q0)

```

## 관련 명령어

- [release\\_force\(time=0\)\(p. 206\)](#)
- [task\\_compliance\\_ctrl\(stx, time\)\(p. 200\)](#)
- [set\\_stiffnessx\(stx, time\)\(p. 201\)](#)
- [release\\_compliance\\_ctrl\(\)\(p. 199\)](#)

### 6.1.5 release\_force(time=0)

#### 기능

힘 제어 목표값을 time 값 동안 0으로 줄이고 작업 공간을 순응 제어로 리턴합니다.

#### 인수

인수명	자료형	기본값	설명
time	float	0	힘을 감소시키는데 소요되는 시간 범위 0~1.0

#### 리턴

값	설명
0	성공
음수값	오류

#### 예외

예외	설명
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

#### 예제

```

1 j0 = posj(0, 0, 90, 0, 90, 0)
2 x0 = posx(0, 0, 0, 0, 0, 0)
3 x1 = posx(0, 500, 700, 0, 180, 0)

```

```

4   x2 = posx(300, 100, 700, 0, 180, 0)
5   x3 = posx(300, 100, 500, 0, 180, 0)
6   set_velx(100,20)
7   set_accx(100,20)
8   movej(j0, vel=10, acc=10)
9   movel(x2)
10  task_compliance_ctrl(stx = [500, 500, 500, 100, 100, 100])
11  fd = [0, 0, 0, 0, 0, 10]
12  fctrl_dir= [0, 0, 1, 0, 0, 1]
13  set_desired_force(fd, dir=fctrl_dir, time=1.0)
14  movel(x3, v=10)
15  release_force(0.5)
16  release_compliance_ctrl()

```

## 관련 명령어

- [set\\_desired\\_force\(fd, dir, time, mod\)](#)(p. 203)
- [task\\_compliance\\_ctrl\(stx, time\)](#)(p. 200)
- [set\\_stiffnessx\(stx, time\)](#)(p. 201)
- [release\\_compliance\\_ctrl\(\)](#)(p. 199)

## 6.1.6 get\_force\_control\_state()

### 기능

컴플라이언스 및 힘 제어 중 상태를 모니터링 할 수 있는 기능입니다.

### 리턴

[singularity, mode, stx, fd, ref]

값	설명
singularity	위험도 : 0 < 1 < 2 0 : 안전 구간 1 : 특이점 위험 구간 1단계 2 : 특이점 위험 구간 2단계
mode	차례대로 x,y,z,rx,ry,rz 6개 mode 정보 0 : 컴플라이언스 제어 1 : 힘 제어 2 : None

값	설명
Stx	차례대로 x,y,z,rx,ry,rz 6개 설정 목표 강성 값 정보
fd	차례대로 x,y,z,rx,ry,rz 6개 설정 목표 힘 값 정보
ref	설정 기준 좌표계 정보 0 : 베이스 좌표계 1 : 툴 좌표계 2 : 월드 좌표계 101 ~ 120 : 유저 좌표계

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 set_ref_coord(DR_BASE)
2 P0 = posj(0, 0, 90, 0, 90, 0)
3 movej(P0, vel=30, acc=60)
4
5 task_compliance_ctrl()
6 set_stiffnessx([500, 500, 500, 100, 100, 100], time=0.5)
7
8 while True:
9     [singularity, mod, stx, fd, ref]=get_force_control_state()
10    #힘 제어 중 모니터링 정보 획득
11    tp_log("s={0}, m={1}, k={2}, f={3}, r={4}".format(singularity,mod,stx,fd,ref))
12    wait(0.5)
13    release_compliance_ctrl()
```

## 관련 명령어

- [set\\_singular\\_handling\\_force\(mode\)\(p. 193\)](#)
- [task\\_compliance\\_ctrl\(stx, time\)\(p. 200\)](#)
- [set\\_stiffnessx\(stx, time\)\(p. 201\)](#)
- [set\\_desired\\_force\(fd, dir, time, mod\)\(p. 203\)](#)
- [release\\_compliance\\_ctrl\(\)\(p. 199\)](#)

### 6.1.7 set\_damping\_factor(damping\_factor, time)

#### 기능

힘 제어 시, 전역 좌표계(set\_ref\_coord() 참조) 기준으로 감쇠값(damping factor)을 설정합니다.

#### 인수

인수명	자료형	기본값	설명
damping_factor	float[6]	[1.0, 1.0, 1.0, 1.0, 1.0, 1.0]	순서: (Translation x, y, z), (Rotation x, y, z) 범위: 0.1 ~ 2.0
time	float	0	감쇠값 변화 시간 [sec] 범위: 0~1.0 * 주어진 시간 동안 linear transition



- 감쇠값을 기본값(1.0)으로 설정할 시 로봇 모델에 따라 설정된 기본 감쇠 제어 이득으로 동작하며, 감쇠값을 변경하면 기본 감쇠 제어 이득에 비례한 값이 힘제어 시 적용됩니다.
- 이 함수를 통해 수정한 감쇠값은 힘제어 시에만 적용 됩니다. 순응 제어 시에는 기본값으로 적용됩니다.
- 본 기능은 M시리즈에 한정하여 지원하고 있습니다. A시리즈와 H시리즈에는 사용할 수 없습니다.



- 감쇠값을 1.0 미만으로 설정 시 접촉면에 따라 로봇에 진동이 발생할 수 있습니다. 신중히 사용하시기 바랍니다.

#### 리턴

값	설명
0	성공

값	설명
음수값	오류

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 # 1. 초기 자세 이동
2 q0 = posj(0.0, 0.0, 90.0, 0.0, 90.0, 0.0)
3 set_velj(30.0)
4 set_accj(60.0)
5 movej(q0)
6
7 # 2. 힘 제어 수행 (Base 좌표계 기준 -z 방향 힘 인가)
8 k_d = [3000.0, 3000.0, 3000.0, 200.0, 200.0, 200.0]
9 task_compliance_ctrl(k_d)
10 force_desired = 20.0
11 f_d = [0.0, 0.0, -force_desired, 0.0, 0.0, 0.0]
12 f_dir = [0, 0, 1, 0, 0, 0]
13 d_f = [1.0, 1.0, 0.8, 1.0, 1.0, 1.0]
14 set_desired_force(f_d, f_dir)
15 set_damping_factor(d_f)
16 wait(2.0)
17 release_force()
18 wait(0.5)
19 release_compliance_ctrl()

```

## 관련 명령어

- [set\\_force\\_factor\(force\\_factor, time\)\(p. 211\)](#)
- [set\\_desired\\_force\(fd, dir, time, mod\)\(p. 203\)](#)
- [release\\_force\(time=0\)\(p. 206\)](#)

## 6.1.8 set\_force\_factor(force\_factor, time)

### 기능

힘 제어 시, 전역 좌표계(set\_ref\_coord() 참조) 기준으로 힘이득값(force factor)을 설정합니다.

### 인수

인수명	자료형	기본값	설명
force_factor	float[6]	[1.0, 1.0, 1.0, 1.0, 1.0, 1.0]	순서: (Translation x, y, z), (Rotation x, y, z) 범위: 0.1 ~ 2.0
time	float	0	힘이득값 변경 시간 [sec] 범위: 0~1.0 * 주어진 시간 동안 linear transition



- 힘이득값은 기본값(1.0)으로 설정시 모델에 따라 설정된 제어 이득으로 동작합니다.
- 힘이득값을 1.0 미만으로 설정 시, 목표 힘 값에 추종하는 시간이 감소하지만 진동이 증가할 수 있습니다.
- 힘이득값을 1.0 초과로 설정 시, 진동을 억제 할 수 있지만 목표 힘 값에 추종하는 시간이 증가합니다.
- 힘이득값 설정은 힘제어 시에만 적용됩니다. 순응 제어 시에는 기본값으로 적용됩니다.
- 본 기능은 M시리즈에 한정하여 지원하고 있습니다. A시리즈와 H시리즈에는 사용할 수 없습니다.



- 힘이득값을 1.0 미만으로 설정 시 접촉면에 따라 로봇에 진동이 발생할 수 있습니다. 신중히 사용하시기 바랍니다.

### 리턴

값	설명
0	성공
음수값	오류

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 # 1. 초기 자세 이동
2 q0 = posj(0.0, 0.0, 90.0, 0.0, 90.0, 0.0)
3 set_velj(30.0)
4 set_accj(60.0)
5 movej(q0)
6
7 # 2. 힘 제어 수행 (Base 좌표계 기준 -z 방향 힘 인가)
8 k_d = [3000.0, 3000.0, 3000.0, 200.0, 200.0, 200.0]
9 task_compliance_ctrl(k_d)
10 force_desired = 20.0
11 f_d = [0.0, 0.0, -force_desired, 0.0, 0.0, 0.0]
12 f_dir = [0, 0, 1, 0, 0, 0]
13 d_f = [1.0, 1.0, 0.8, 1.0, 1.0, 1.0]
14 f_f = [1.0, 1.0, 1.2, 1.0, 1.0, 1.0]
15 set_desired_force(f_d, f_dir)
16 set_damping_factor(d_f)
17 set_force_factor(f_f)
18 wait(2.0)
19 release_force()
20 wait(0.5)
21 release_compliance_ctrl()

```

## 관련 명령어

- [set\\_damping\\_factor\(damping\\_factor, time\)\(p. 209\)](#)
- [set\\_desired\\_force\(fd, dir, time, mod\)\(p. 203\)](#)
- [release\\_force\(time=0\)\(p. 206\)](#)

## 6.2 사용자 편의 기능

### 6.2.1 parallel\_axis(x1, x2, x3, axis, ref)

#### 기능

입력된 기준좌표계(ref) 기준의 3개의 포즈(x1,x2,x3)가 이루는 평면의 normal vector(get\_normal(x1, x2, x3) 참조) 방향에 Tool좌표계의 지정축(axis)의 방향을 일치시킵니다. 이때 로봇 TCP 위치는 현재 위치를 유지합니다

#### 인수

인수명	자료형	기본값	설명
x1	posx	-	posx 또는 position list
	list (float[6])		
x2	posx	-	posx 또는 position list
	list (float[6])		
x3	posx	-	posx 또는 position list
	list (float[6])		
axis	int	-	axis <ul style="list-style-type: none"> <li>• DR_AXIS_X: x축</li> <li>• DR_AXIS_Y: y축</li> <li>• DR_AXIS_Z: z축</li> </ul>
ref	int	DR_BASE	reference coordinate <ul style="list-style-type: none"> <li>• DR_BASE: base coordinate</li> <li>• DR_WORLD: world coordinate</li> <li>• user coordinate: 사용자 정의</li> </ul>

#### 리턴

값	설명
0	성공

값	설명
음수값	오류

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 x0 = posx(0, 0, 90, 0, 90, 0)
2 movej(x0)
3 x1 = posx(0, 500, 700, 30, 0, 90)
4 x2 = posx(500, 0, 700, 0, 0, 45)
5 x3 = posx(300, 100, 500, 45, 0, 45)
6 parallel_axis(x1, x2, x3, DR_AXIS_X, DR_WORLD)
7 #WORLD좌표계 기준 x1,x2,x3로 이루어진 평면에 수직인 방향에 툴 X축을 일치

```

## 관련 명령어

- [get\\_normal\(x1, x2, x3\)\(p. 330\)](#)
- [parallel\\_axis\(vect, axis, ref\)\(p. 214\)](#)
- [align\\_axis\(vect, pos, axis, ref\)\(p. 218\)](#)
- [align\\_axis\(x1, x2, x3, pos, axis, ref\)\(p. 216\)](#)

## 6.2.2 parallel\_axis(vect, axis, ref)

### 기능

입력된 기준좌표계(ref) 기준의 벡터(vect) 방향에 Tool좌표계의 지정축(axis)의 방향을 일치시킵니다. 이때 로봇 TCP 위치는 현재 위치를 유지합니다.

## 인수

인수명	자료형	기본값	설명
vect	list (float[3])	-	vector
axis	int	-	axis <ul style="list-style-type: none"> <li>• DR_AXIS_X: x축</li> <li>• DR_AXIS_Y: y축</li> <li>• DR_AXIS_Z: z축</li> </ul>
ref	int	DR_BASE	reference coordinate <ul style="list-style-type: none"> <li>• DR_BASE: base coordinate</li> <li>• DR_WORLD: world coordinate</li> <li>• user coordinate: 사용자 정의</li> </ul>

## 리턴

값	설명
0	성공
음수값	오류

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1  x0 = posx(0, 0, 90, 0, 90, 0)
2  movej(x0)
3  parallel_axis([1000, 700, 300], DR_AXIS_X, DR_WORLD)

```

4

# WORLD 좌표계 기준 [1000,700,300] vector방향으로 툴의 X축을 일치

## 관련 명령어

- movej()(p. 57)
- parallel\_axis(x1, x2, x3, axis, ref)(p. 213)
- align\_axis(vect, pos, axis, ref)(p. 218)
- align\_axis(x1, x2, x3, pos, axis, ref)(p. 216)

### 6.2.3 align\_axis(x1, x2, x3, pos, axis, ref)

#### 기능

입력된 기준좌표계(ref) 기준의 3개의 포즈(x1,x2,x3)가 이루는 평면의 normal vector(get\_normal(x1, x2, x3) 참조) 방향에 Tool좌표계의 지정축(axis)의 방향을 일치시킵니다. 이때 로봇 TCP 위치는 pos 위치로 이동합니다.

#### 인수

인수명	자료형	기본값	설명
x1	posx	-	posx 또는 position list
	list (float[6])		
x2	posx	-	posx 또는 position list
	list (float[6])		
x3	posx	-	posx 또는 position list
	list (float[6])		
pos	posx	-	posx 또는 position list
	list (float[6])		
axis	int	-	axis <ul style="list-style-type: none"> <li>• DR_AXIS_X: x축</li> <li>• DR_AXIS_Y: y축</li> <li>• DR_AXIS_Z: z축</li> </ul>

인수명	자료형	기본값	설명
ref	int	DR_BASE	<p>reference coordinate</p> <ul style="list-style-type: none"> <li>• DR_BASE: base coordinate</li> <li>• DR_WORLD: world coordinate</li> <li>• user coordinate: 사용자 정의</li> </ul>

## 리턴

값	설명
0	성공
음수값	오류

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 p0 = posj(0,0,45,0,90,0)
2 movej(p0, v=30, a=30)
3
4 x1 = posx(0, 500, 700, 30, 0, 0)
5 x2 = posx(500, 0, 700, 0, 0, 0)
6 x3 = posx(300, 100, 500, 0, 0, 0)
7 pos = posx(400, 400, 500, 0, 0, 0)
8 align_axis(x1, x2, x3, pos, DR_AXIS_X, DR_BASE)
9 #BASE좌표계 기준 x1,x2,x3로 이루어진 평면에 수직인 방향에 툴 X축 방향을, pos에 위치를 일치

```

## 관련 명령어

- [movej\(\)\(p. 57\)](#)
- [get\\_normal\(x1, x2, x3\)\(p. 330\)](#)
- [align\\_axis\(vect, pos, axis, ref\)\(p. 218\)](#)
- [parallel\\_axis\(vect, axis, ref\)\(p. 214\)](#)
- [parallel\\_axis\(x1, x2, x3, axis, ref\)\(p. 213\)](#)

### 6.2.4 align\_axis(vect, pos, axis, ref)

#### 기능

입력된 기준좌표계(ref) 기준의 벡터(vect) 방향에 Tool좌표계의 지정축(axis)의 방향을 일치시킵니다. 이때 로봇 TCP 위치는 pos 위치로 이동시킵니다.

#### 인수

인수명	자료형	기본값	설명
vect	list (float[3])	-	vector
pos	posx	-	posx 또는 position list
	list (float[6])		
axis	int	-	axis <ul style="list-style-type: none"> <li>• DR_AXIS_X: x축</li> <li>• DR_AXIS_Y: y축</li> <li>• DR_AXIS_Z: z축</li> </ul>
ref	int	DR_BASE	reference coordinate <ul style="list-style-type: none"> <li>• DR_BASE: base coordinate</li> <li>• DR_WORLD: world coordinate</li> <li>• user coordinate : 사용자 정의</li> </ul>

#### 리턴

값	설명
0	성공
음수값	오류

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 p0 = posj(0,0,45,0,90,0)
2 movej(p0, v=30, a=30)
3
4 vect = [10,20,30]
5 pos = posx(100, 500, 700, 45, 0, 0)
6 align_axis(vect, pos, DR_AXIS_X)
7 align_axis(vect, pos, DR_AXIS_X, DR_WORLD)
8 #WORLD좌표계 기준 [10,20,30]벡터 방향에 틀 X축 방향을, pos에 위치를 일치

```

## 관련 명령어

- movej()(p. 57)
- align\_axis(x1, x2, x3, pos, axis, ref)(p. 216)
- parallel\_axis(vect, axis, ref)(p. 214)
- parallel\_axis(x1, x2, x3, axis, ref)(p. 213)

## 6.2.5 is\_done\_bolt\_tightening(m=0, timeout=0, axis=None)

### 기능

틀의 조임 토크를 모니터링하여 주어진 시간 내에 설정된 토크(m)에 도달한 경우는 True를 리턴하고, 주어진 시간을 초과한 경우에는 False를 리턴합니다.

### 인수

인수명	자료형	기본값	설명
m	float	0	Target torque

인수명	자료형	기본값	설명
timeout	float	0	Monitoring duration [sec]
axis	int	-	axis <ul style="list-style-type: none"> <li>• DR_AXIS_X: x축</li> <li>• DR_AXIS_Y: y축</li> <li>• DR_AXIS_Z: z축</li> </ul>

## 리턴

값	설명
0	성공
음수값	오류

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 p0 = posj(0,0,90,0,90,0)
2 movej(p0, v=30, a=30)
3
4 task_compliance_ctrl()
5 xd = posx(559, 34.5, 651.5, 0, 180.0, 60)
6 amovel(xd, vel=50, acc=50) # 볼트 조이는 모션
7
8 res = is_done_bolt_tightening(10, 5, DR_AXIS_Z)
9 # 5초 내에 10Nm을 조임 토크에 도달한 경우는 True,
10 # 그렇지 않은 경우는 False를 Return 하십시오.
11 if res==True:
12     # some action comes here for the case that bolt tightening is done
13     x=1
14 else:
15     # some action comes here for the case that it fails

```

16 x=2

## 관련 명령어

- movej()(p. 57)
- amovel()(p. 103)

### 6.2.6 calc\_coord(x1, x2, x3, x4, ref, mod, ori\_type\_out)

#### 기능

지정한 좌표계(ref) 기준의 최대 4개의 입력점(x1~x4) 및 입력 모드(mod)를 기반으로 새로운 직교 좌표계를 계산할 수 있습니다. 여기서 입력 모드는 입력점의 개수가 2개인 경우에만 유효합니다.

입력점의 개수가 1개인 경우, x1의 위치와 회전으로 좌표계가 계산됩니다.

입력점의 개수가 2개인 경우 입력모드가 0일 때, x1에서 x2로 향하는 벡터가 x방향, x축에 직교하는 평면에 투영된 현재의 Tool-z방향이 사용자 좌표계의 z방향으로 정의되며 x1의 위치가 원점이 되도록 좌표계가 계산됩니다.

입력점의 개수가 2개인 경우 입력모드가 1일 때, x1에서 x2로 향하는 벡터가 x방향, x축에 직교하는 평면에 투영된 x1의 z축방향이 사용자 좌표계의 z방향으로 정의되며 x1의 위치가 원점이 되도록 좌표계가 계산됩니다.

입력점의 개수가 3개인 경우, x1에서 x2로 향하는 벡터가 x방향으로 정의되며, x1에서 x3으로 향하는 벡터를 v라고 하였을 경우 z방향은 오른손법칙에 따라 x방향 벡터 곱하기 v로 정의되며, x1의 위치가 원점이 되도록 좌표계가 계산됩니다.

입력점의 개수가 4개인 경우, 입력점의 개수가 3개인 경우와 축의 방향은 동일하며 원점의 위치가 x4의 위치가 되도록 좌표계가 계산됩니다

#### 인수

인수명	자료형	기본값	설명
x1, x2, x3, x4	posx list (float[6])	-	posx 또는 position list
ref	int	-	reference coordinate <ul style="list-style-type: none"> <li>• DR_BASE: base coordinate</li> <li>• DR_WORLD: world coordinate</li> </ul>
mod	int	-	입력 모드 (입력점개수가 2개인 경우에만 유효함) <ul style="list-style-type: none"> <li>• 0: 현재 Tool-z방향 기준으로 사용자 좌표계의 z 방향 정의</li> <li>• 1: x1의 z방향 기준으로 사용자 좌표계의 z방향 정의</li> </ul>

인수명	자료형	기본값	설명
ori_type_out	int	None	<p>orientation type</p> <ul style="list-style-type: none"> <li>• None: pos 인자의 orientation type을 따름.</li> <li>• DR_ELR_ZYX: Euler Angles(z-y'-z'', in degrees)</li> <li>• DR_ELR_ZYX: Euler Angles(z-y'-x'', in degrees)</li> <li>• DR_ELR_XYZ: Euler Angles(x-y'-z'', in degrees)</li> <li>• DR_FIX_XYZ: Fixed Angles(x-y-z, in degrees)</li> <li>• DR_ROTVEC: 3D rotation vector (angle/axis representation)</li> <li>• DR_QUAT: unit quaternion(x, y, z, w)</li> </ul>

## 리턴

값	설명
posx	Coordinate 계산 성공 설정된 Coordinate의 위치정보

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1  pos1 = posx(500, 30, 500, 0, 0, 0)
2  pos2 = posx(400, 30, 500, 0, 0, 0)
3  pos3 = posx(500, 30, 600, 45, 180, 45)
4  pos4 = posx(500, -30, 600, 0, 180, 0)
5  pose_user1 = calc_coord(pos1, ref=DR_BASE, mod=0, ori_type_out=DR_ELR_ZYX
   )
6  pose_user21 = calc_coord(pos1, pos2, ref=DR_WORLD, mod=0)
7  %% 현재 Tool-z방향 기준으로 사용자 좌표계의 z방향 정의
8  pose_user22 = calc_coord(pos1, pos2, ref=DR_BASE, mod=1)
9  %% pos1의 z방향 기준으로 사용자 좌표계의 z방향 정의
10 pose_user3 = calc_coord(pos1, pos2, pos3, ref=DR_BASE, mod=0)

```

```

11  pose_user4 = calc_coord(pos1, pos2, pos3, pos4, ref=DR_WORLD, mod=0)
12  ucart1 = set_user_cart_coord(pose_user1, ref=DR_BASE)
13  ucart2 = set_user_cart_coord(pose_user2, ref=DR_WORLD)

```

## 관련 명령어

- [set\\_user\\_cart\\_coord\(pos, ref\)](#)(p. 223)
- [set\\_user\\_cart\\_coord\(u1, v1, pos, ref\)](#)(p. 226)
- [set\\_user\\_cart\\_coord\(x1, x2, x3, pos, ref\)](#)(p. 224)

## 6.2.7 set\_user\_cart\_coord(pos, ref)

### 기능

기준 좌표계(ref) 기반의 새로운 사용자좌표계를 설정할 수 있습니다. Workcell Item에서 설정한 좌표계를 포함하여 총 100개의 사용자좌표계를 설정할 수 있으며, 100개가 넘어가면 새로운 직교 좌표계를 설정할 수 없습니다. 명령어를 통해 설정한 사용자좌표계는 프로그램 실행 종료 시 삭제되므로, 사용자좌표계 정보를 유지하려면 Workcell Item에서 사용자좌표계를 설정하세요.

### 인수

인수명	자료형	기본값	설명
pos	posx	-	사용자좌표계 정보 (위치 및 방향)
	list (float[6])		
ref	int	-	reference coordinate <ul style="list-style-type: none"> <li>• DR_BASE: base coordinate</li> <li>• DR_WORLD: world coordinate</li> </ul>

### 리턴

값	설명
양의 정수	Coordinate 설정 성공 설정된 Coordinate ID (101 ~ 120)
-1	Coordinate 설정 실패

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 pos1 = posx(10, 20, 30, 0, 0, 0)
2 pos2 = posx(30, 50, 70, 45, 180, 45)
3 user_id1 = set_user_cart_coord(pos1, ref=DR_BASE)
4 user_id2 = set_user_cart_coord(pos2, ref=DR_WORLD)

```

## 관련 명령어

- [set\\_ref\\_coord\(coord\)\(p. 53\)](#)

## 6.2.8 set\_user\_cart\_coord(x1, x2, x3, pos, ref)

### 기능

사용자가 입력좌표계(ref) 기준의 포즈 x1, x2, x3를 사용하여 새로운 직교 좌표계를 설정할 수 있습니다.<sup>1)</sup> x1x2의 단위 벡터를 ux, x1x2로부터 x3까지 최단거리로 잇는 vector의 단위벡터를 uy로 하여, ux, uy, uz를 각 축의 방향 벡터, 원점은 입력좌표계(ref) 기준의 pos에 위치한 직교 좌표계를 생성합니다. Workcell Item에서 설정한 좌표계를 포함하여 총 100개의 사용자좌표계를 설정할 수 있으며, 100개가 넘어가면 새로운 직교 좌표계를 설정할 수 없습니다. 명령어를 통해 설정한 사용자좌표계는 프로그램 실행 종료 시 삭제되므로, 사용자좌표계 정보를 유지하려면 Workcell Item에서 사용자좌표계를 설정하세요.

<sup>1)</sup>M2.0.2 이전 버전에서는 x2x1의 단위 벡터를 ux로 사용

### 인수

인수명	자료형	기본값	설명
x1	Posx	-	posx 또는 position list

인수명	자료형	기본값	설명
	list (float[6])		
x2	Posx	-	posx 또는 position list
	list (float[6])		
x3	Posx	-	posx 또는 position list
	list (float[6])		
pos	Posx	-	posx 또는 position list
	list (float[6])		
ref	int	DR_BASE	<p>reference coordinate</p> <ul style="list-style-type: none"> <li>• DR_BASE: base coordinate</li> <li>• DR_WORLD: world coordinate</li> </ul>

## 리턴

값	설명
양의 정수	Coordinate 설정 성공 설정된 Coordinate ID (101 ~ 200)
-1	Coordinate 설정 실패

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 x1 = posx(0, 500, 700, 0, 0, 0) # Euler angle은 계산시 무시
2 x2 = posx(500, 0, 700, 0, 0, 0)
3 x3 = posx(300, 100, 500, 0, 0, 0)
4 x4 = posx(300, 110, 510, 0, 0, 0)
5 pos = posx(10, 20, 30, 0, 0, 0)
6 user_tc1 = set_user_cart_coord(x1, x2, x3, pos, ref=DR_BASE)
7 user_tc2 = set_user_cart_coord(x2, x3, x4, pos, ref=DR_WORLD)

```

## 관련 명령어

- [set\\_ref\\_coord\(coord\)](#)(p. 53)

### 6.2.9 set\_user\_cart\_coord(u1, v1, pos, ref)

#### 기능

사용자가 입력좌표계(ref) 기준의 벡터 u1과 v1를 사용하여 새로운 직교 좌표계를 설정할 수 있습니다. 직교 좌표계의 원점은 입력좌표계(ref) 기준의 pos에 위치하고, x축/y축 basis는 vector u1과 v1에 주어집니다. 나머지 방향은 u1 x v1에 의해 정해집니다. u1과 v1이 orthogonal 하지 않은 경우, u1과 v1이 span 하는 평면상에 u1과 수직인 v1'를 y 축의 방향 vector로 설정합니다. Workcell Item에서 설정한 좌표계를 포함하여 총 100개의 사용자좌표계를 설정할 수 있으며, 100개가 넘어가면 새로운 직교 좌표계를 설정할 수 없습니다. 명령어를 통해 설정한 사용자좌표계는 프로그램 실행 종료 시 삭제되므로, 사용자좌표계 정보를 유지하려면 Workcell Item에서 사용자좌표계를 설정하세요.

#### 인수

인수명	자료형	기본값	설명
u1	float[3]	-	X축 단위벡터
v1	float[3]	-	y축 단위벡터
pos	posx list (float[6])	-	posx 또는 position list
ref	int	DR_BASE	reference coordinate <ul style="list-style-type: none"> <li>• DR_BASE: base coordinate</li> <li>• DR_WORLD: world coordinate</li> </ul>

## 리턴

값	설명
양의 정수	Coordinate 설정 성공 설정된 Coordinate ID (101 ~ 200)
-1	Coordinate 설정 실패

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 u1 = [1, 1, 0]
2 v1 = [-1, 1, 0]
3 pos = posx(10, 20, 30, 0, 0, 0)
4 user_tc1 = set_user_cart_coord(u1, v1, pos)
5 user_tc2 = set_user_cart_coord(u1, v1, pos, ref=DR_WORLD)

```

## 관련 명령어

- [set\\_ref\\_coord\(coord\)\(p. 53\)](#)

### 6.2.10 overwrite\_user\_cart\_coord(id, pos, ref, apply\_mod)

## 기능

요청하는 ID(id)의 사용자 좌표계의 좌표계 위치(pos), 기준 좌표계(ref) 정보를 변경합니다.

## 인수

인수명	자료형	기본값	설명
id	int	-	사용자 좌표계 ID
pos	posx list (float[6])	-	사용자좌표계 정보 (위치 및 방향)
ref	int	DR_BASE	reference coordinate <ul style="list-style-type: none"> <li>• DR_BASE: base coordinate</li> <li>• DR_WORLD: world coordinate</li> </ul>
apply_mod	int	DR_TEMPORARY	DR_TEMPORARY(0) : 프로그램 실행 중에만 유효 DR_PERMANENT(1) : 프로그램 종료 후에도 유효

**i 정보**

apply\_mod가 0일 경우에는 프로그램 실행 시에만 유저 좌표계가 변경되어 유지되며, 1일 경우에는 상위 제어기의 저장된 유저 좌표계 정보 자체가 변경됩니다.

## 리턴

값	설명
양의 정수	변경된 Coordinate의 ID, 참조 기준 및 위치정보
-1	Coordinate 계산 실패

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1  pose_user1 = posx(30, 40, 50, 0, 0, 0)
2  id_user = set_user_cart_coord(pose_user1, ref=DR_BASE)
3  pose_user2 = posx(100, 150, 200, 45, 180, 0)
4  overwrite_user_cart_coord(id_user, pose_user2, ref=DR_BASE)
5  pose_user3 = posx(100, 150, 200, 45, 180, 0)
6  overwrite_user_cart_coord(id_user, pose_user3, ref=DR_BASE, apply_mod=DR_P
    ERMANENT)

```

## 관련 명령어

- [set\\_user\\_cart\\_coord\(pos, ref\)](#)(p. 223)
- [set\\_user\\_cart\\_coord\(u1, v1, pos, ref\)](#)(p. 226)
- [set\\_user\\_cart\\_coord\(x1, x2, x3, pos, ref\)](#)(p. 224)

### 6.2.11 get\_user\_cart\_coord(id, ori\_type)

#### 기능

해당하는 ID(id)의 사용자 좌표계의 정보인 참조 기준 및 위치정보를 조회합니다.

- 리턴하고자 하는 posx 값의 ori\_type을 인수로 추가

#### 인수

인수명	자료형	기본값	설명
id	int	-	사용자 좌표계 ID
ori_type	int	DR_ELR_ZYZ	<p>orientation type</p> <ul style="list-style-type: none"> <li>DR_ELR_ZYZ: Euler Angles(z-y'-z'', in degrees)</li> <li>DR_ELR_ZYX: Euler Angles(z-y'-x'', in degrees)</li> <li>DR_ELR_XYZ: Euler Angles(x-y'-z'', in degrees)</li> <li>DR_FIX_XYZ: Fixed Angles(x-y-z, in degrees)</li> <li>DR_ROTVEC: 3D rotation vector (angle/axis representation)</li> <li>DR_QUAT: unit quaternion(x, y, z, w)</li> </ul>

## 리턴

값	설명
posx	조회하고자 하는 Coordinate의 위치정보
ref	조회하고자 하는 Coordinate의 부모 좌표계

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 pose_user1 = posx(10, 20, 30, 0, 0, 0)
2 id_user = set_user_cart_coord(pose_user1, ref=DR_BASE)
3 pose, ref = get_user_cart_coord(id_user)
4 pose_quat, ref = get_user_cart_coord(id_user, DR_QUAT)

```

## 관련 명령어

- [set\\_user\\_cart\\_coord\(pos, ref\)\(p. 223\)](#)
- [set\\_user\\_cart\\_coord\(u1, v1, pos, ref\)\(p. 226\)](#)
- [set\\_user\\_cart\\_coord\(x1, x2, x3, pos, ref\)\(p. 224\)](#)

## 6.2.12 check\_position\_condition(axis, min, max, ref, mod, pos)

### 기능

주어진 위치 상태를 확인합니다. while 또는 if 조건과 함께 사용하여 해당 조건을 반복 확인할 수 있습니다. axis, pos는 입력좌표계(ref) 기준의 축방향 및 포즈입니다.

입력좌표계(ref)가 DR\_TOOL 인 경우 입력위치(pos)는 BASE좌표계 기준의 값을 입력하여야 합니다.

## 인수

인수명	자료형	기본값	설명
axis	int	-	axis <ul style="list-style-type: none"> <li>• DR_AXIS_X: x축</li> <li>• DR_AXIS_Y: y축</li> <li>• DR_AXIS_Z: z축</li> </ul>
min	float	DR_COND_NONE	최소값
max	float	DR_COND_NONE	최대값
ref	int	None	reference coordinate <ul style="list-style-type: none"> <li>• DR_BASE : base coordinate</li> <li>• DR_WORLD : world coordinate</li> <li>• DR_TOOL : tool coordinate</li> <li>• user coordinate: 사용자 정의</li> </ul>
mod	int	DR_MV_MOD_ABS	이동 기준 <ul style="list-style-type: none"> <li>• DR_MV_MOD_ABS: 절대</li> <li>• DR_MV_MOD_REL: 상대</li> </ul>
pos	posx list (float[6])	-	posx 또는 position list

### ● 알아두기

- mod가 DR\_MV\_MOD\_ABS인 경우는 절대 위치 기준으로 확인합니다.
- mod가 DR\_MV\_MOD\_REL인 경우는 pos 위치 기준으로 확인합니다.
- pos는 mod가 DR\_MV\_MOD\_REL인 경우에만 의미가 있습니다

## 리턴

값	설명
True	조건이 참

값	설명
False	조건이 거짓

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 CON1= check_position_condition(DR_AXIS_X, min=-5, max=0, ref=DR_WORLD)
2 CON2= check_position_condition(DR_AXIS_Y, max=700)
3 CON3= check_position_condition(DR_AXIS_Z, min=-10, max=-5) # -10≤z≤-5
4 CON4= check_position_condition(DR_AXIS_Z, min=30) # 30≤z
5
6 CON5= check_position_condition(DR_AXIS_Z,min=-10,max=-5, ref=DR_BASE) #
-10≤z≤-5
7
8 CON6= check_position_condition(DR_AXIS_Z,min=-10,max=-5, mod=DR_MV_MOD_ABS
) # -10≤z≤-5
9
10 posx1 = posx(400, 500, 800, 0, 180,0)
11 CON7= check_position_condition(DR_AXIS_Z,min=-10,max=-5,mod =
DR_MV_MOD_REL, pos=posx1) # posx1_(z)-10≤z≤ posx1_(z)-5

```

## 관련 명령어

- [check\\_force\\_condition\(axis, min, max, ref\)\(p. 233\)](#)
- [check\\_orientation\\_condition\(axis, min, max, ref, mod\)\(p. 234\)](#)
- [check\\_orientation\\_condition\(axis, min, max, ref, mod, pos\)\(p. 237\)](#)
- [set\\_ref\\_coord\(coord\)\(p. 53\)](#)

## 6.2.13 check\_force\_condition(axis, min, max, ref)

### 기능

주어진 힘 상태를 확인합니다. 단, 힘의 방향은 고려하지 않고 크기로만 비교합니다. while 또는 if 조건과 함께 사용하여 해당 조건을 반복 확인할 수 있습니다. 힘, <sup>1)</sup>모멘트 측정 시 axis는 입력좌표계(ref) 기준의 축방향입니다.

<sup>1)</sup>V2.8버전 이전에서 모멘트 측정 시 axis는 툴 좌표계 기준의 축방향입니다.

### 인수

인수명	자료형	기본값	설명
axis	int	-	axis <ul style="list-style-type: none"> <li>• DR_AXIS_X: x축</li> <li>• DR_AXIS_Y: y축</li> <li>• DR_AXIS_Z: z축</li> <li>• DR_AXIS_A: x축 회전</li> <li>• DR_AXIS_B: y축 회전</li> <li>• DR_AXIS_C: z축 회전</li> </ul>
min	float	DR_COND_NONE	최소값 ( $\text{min} \geq 0$ )
max	float	DR_COND_NONE	최대값 ( $\text{max} \geq 0$ )
ref	int	None	reference coordinate <ul style="list-style-type: none"> <li>• DR_BASE : base coordinate</li> <li>• DR_WORLD : world coordinate</li> <li>• DR_TOOL : tool coordinate</li> <li>• user coordinate: 사용자 정의</li> </ul>

### 리턴

값	설명
True	조건이 참
False	조건이 거짓

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 fcon1 = check_force_condition(DR_AXIS_Z, min=5, max=10, ref=DR_WORLD)
2 # 5 ≤f_z≤10
3
4 while (fcon1):
5   fcon2 = check_force_condition(DR_AXIS_C, min=30) # 30≤m_z
6   pcon1 = check_position_condition(DR_AXIS_X, min=0, max=0.1) # 0≤x≤0.1
7
8   if (fcon2 and pcon1):
9     break

```

## 관련 명령어

- `check_position_condition(axis, min, max, ref, mod, pos)`(p. 230)
- `check_orientation_condition(axis, min, max, ref, mod)`(p. 234)
- `check_orientation_condition(axis, min, max, ref, mod, pos)`(p. 237)
- `set_ref_coord(coord)`(p. 53)

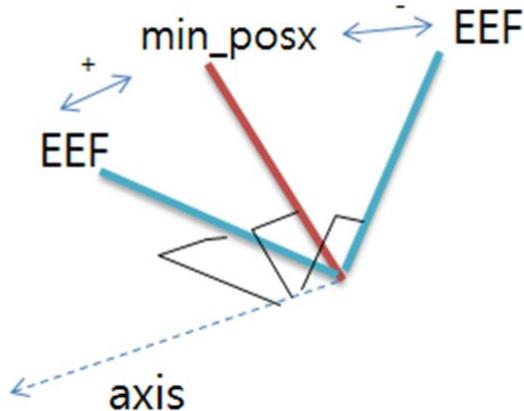
## 6.2.14 `check_orientation_condition(axis, min, max, ref, mod)`

### 기능

현재 로봇 엔드이펙터의 자세 정보와 주어진 위치 자세 간 차이의 상태를 확인합니다. 현재 자세와 주어진 자세 간의 차이는 알고리즘 내부에서 회전행렬로 변환되어 “AngleAxis” 기법으로 차이 값(rad 단위)을 리턴합니다. 차이가 + 값이면 true를, - 값이면 false를 리턴합니다. 현재 자세를 기준으로, 주어진 position보다 차이가 +인지 -인지 확인할 때 사용합니다. 사용 예로, 직접 교시 `position`을 이용하여 현재 위치와 차이가 + 방향인지, - 방향인지를 판단한 후 `orientation limit`에 대한 조건을 만들 수 있습니다. 또한 while 또는 if 조건과 함께 사용하여 해당 조건을 반복 확인 할 수 있습니다.

- Min만 설정 시: 차이가 +이면 True, -이면 False

- Min, Max 설정 시: min과의 차이가 -이고, max 차이가 +이면 True, 반대면 False
- Max만 설정 시: max 차이가 -이면 True, +이면 False



## 인수

인수명	자료형	기본값	설명
axis	int	-	axis <ul style="list-style-type: none"> <li>DR_AXIS_A: x축 회전</li> <li>DR_AXIS_B: y축 회전</li> <li>DR_AXIS_C: z축 회전</li> </ul>
min	posx	-	posx 또는 position list
	list (float[6])		
max	posx	-	posx 또는 position list
	list (float[6])		
ref	int	None	reference coordinate <ul style="list-style-type: none"> <li>DR_BASE : base coordinate</li> <li>DR_WORLD : world coordinate</li> <li>DR_TOOL : tool coordinate</li> <li>user coordinate: 사용자 정의</li> </ul>
mod	int	DR_MV_MOD_AB S	이동 기준 <ul style="list-style-type: none"> <li>DR_MV_MOD_ABS: 절대</li> </ul>

## 리턴

값	설명
True	조건이 참
False	조건이 거짓

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 posx1 = posx(400,500,800,0,180,30)
2 posx2 = posx(400,500,500,0,180,60)
3
4 CON1= check_orientation_condition(DR_AXIS_C, min=posx1, max= posx2)
5 # 현재 Task 좌표가 posxc = posx(400,500,500,0,180,40)인 경우
6 # posx1 Rz=30 < posxc Rz=40 < posx2 Rz=60이므로 CON1=True 값이 됩니다.
7
8 CON2= check_orientation_condition(DR_AXIS_C, min=posx1)
9 # 현재 Task 좌표가 posxc = posx(400,500,500,0,180,15)인 경우
10 # posx1 Rz= 30 > posxc Rz=15이므로 CON2=False 값이 됩니다.
11
12 CON3= check_orientation_condition(DR_AXIS_C, max= posx2)
13 # 현재 Task 좌표가 posxc = posx(400,500,500,0,180,75)인 경우
14 # posx1 Rz= 75 > posxc Rz = 60이므로 CON2=False 값이 됩니다.

```

## 관련 명령어

- [check\\_position\\_condition\(axis, min, max, ref, mod, pos\)\(p. 230\)](#)
- [check\\_force\\_condition\(axis, min, max, ref\)\(p. 233\)](#)

- `check_orientation_condition(axis, min, max, ref, mod)`<sup>24</sup>
- `check_orientation_condition(axis, min, max, ref, mod, pos)`(p. 237)
- `set_ref_coord(coord)`(p. 53)

## 6.2.15 `check_orientation_condition(axis, min, max, ref, mod, pos)`

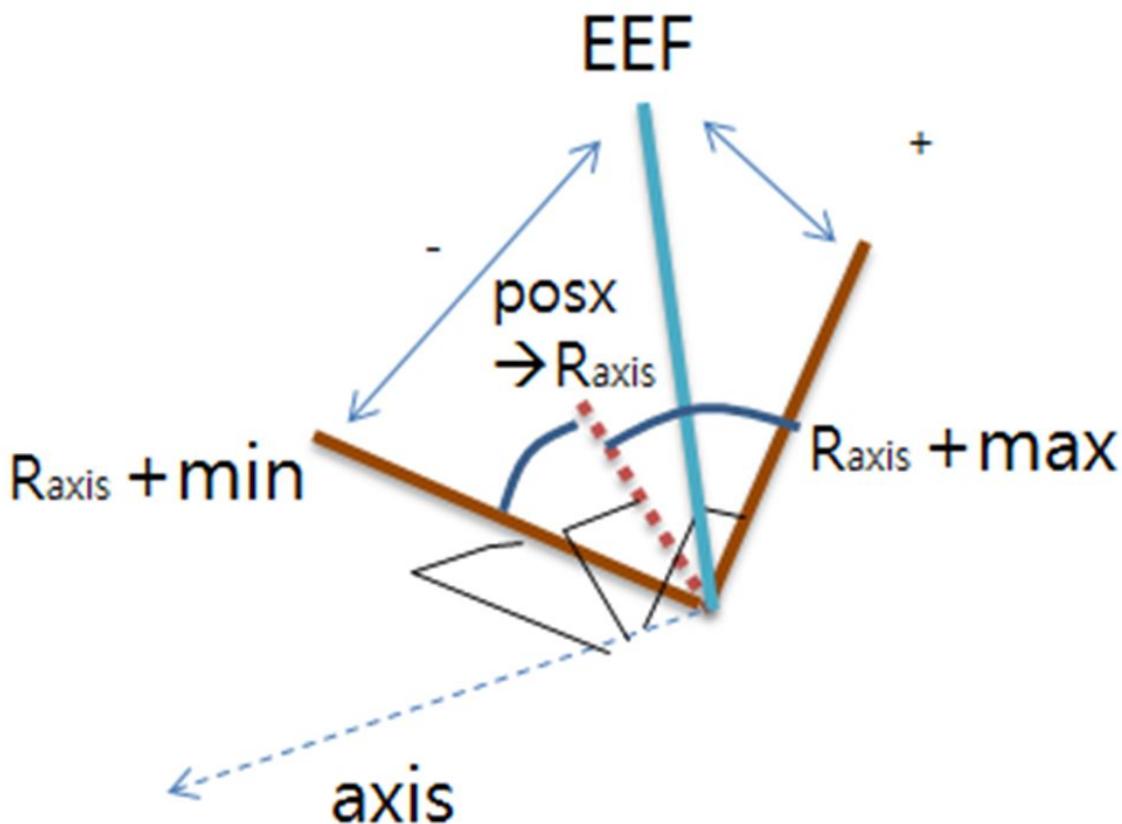
### 기능

현재 로봇 엔드이펙터의 자세와 회전각 범위 차이에 대한 상태를 확인합니다. 현재 자세와 회전각 범위에 대한 차이는 알고리즘 내부에서 회전행렬로 변환되어 “AngleAxis” 기법으로 차이 값(rad 단위)을 리턴합니다. 차이가 + 값이면 true를, -값이면 false를 리턴합니다. 현재 자세를 기준으로, 주어진 position과 회전각 범위 차이가 +인지 -인지 확인 할 때 사용합니다. 사용 예로, 어떤 기준이 되는 position에서 min, max로 회전각 범위를 설정하여, 현재 위치와 차이 가 + 방향인지, - 방향인지 판단한 후 orientation limit에 대한 조건을 만들 수 있습니다. 또한 while 또는 if 조건과 함께 사용하여 해당 조건을 반복 확인할 수 있습니다.

- Min만 설정 시: 차이가 +이면 True, -이면 False
- Min, Max 설정 시: min과의 차이가 -이고, max 차이가 +이면 True, 그 반대이면 False
- Max만 설정 시: max 차이가 -이면 True, +이면 False

---

<sup>24</sup> [https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/5014377/check\\_orientation\\_condition+axis+min+max+ref+mod](https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/5014377/check_orientation_condition+axis+min+max+ref+mod)



### **i 알아두기**

회전각 범위: 주어진 position에서 설정된 axis를 기준으로, 상대적인 각도 범위(min, max)를 말합니다. 인자 ref에 따라 주어진 position의 기준 좌표계가 정해집니다.

### 인수

인수명	자료형	기본값	설명
axis	int	-	axis <ul style="list-style-type: none"> <li>• DR_AXIS_X: x축 회전</li> <li>• DR_AXIS_Y: y축 회전</li> <li>• DR_AXIS_Z: z축 회전</li> </ul>
min	float	DR_COND_NO_NE	최솟값
max	float	DR_COND_NO_NE	최댓값

인수명	자료형	기본값	설명
ref	int	None	reference coordinate • DR_BASE : base coordinate • DR_WORLD : world coordinate • DR_TOOL : tool coordinate • user coordinate: 사용자 정의
mod	int	DR_MV_MOD_REL	이동 기준 • DR_MV_MOD_REL: 상대
pos	posx	-	posx 또는 position list
	list (float[6])		

## 리턴

값	설명
True	조건이 참
False	조건이 거짓

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 posx1 = posx(400,500,800,0,180,15)
2 CON1= check_orientation_condition(DR_AXIS_C, min=-5, mod=DR_MV_MOD_REL,
3 pos=posx1, DR_WORLD)
4 # 현재 Task 좌표가 posxc = posx(400,500,500,0,180,40) 인 경우
# posx1 Rz=15 - (min=5) < posxc Rz=40 이므로 CON1=True 값이 됨.

```

```

5   CON1= check_orientation_condition(DR_AXIS_C, max=5, mod=DR_MV_MOD_REL, pos=posx1)
6
7   # 현재 Task 좌표가 posxc = posx(400,500,500,0,180,40) 인 경우
8   # posxc Rz=40 > posx1 Rz=15 + (max=5) 이므로 CON1=False 값이 됨

```

## 관련 명령어

- `check_position_condition(axis, min, max, ref, mod, pos)`(p. 230)
- `check_force_condition(axis, min, max, ref)`(p. 233)
- `check_orientation_condition(axis, min, max, ref, mod)`(p. 234)
- `check_orientation_condition(axis, min, max, ref, mod, pos)`<sup>25</sup>
- `set_ref_coord(coord)`(p. 53)

## 6.2.16 coord\_transform(pose\_in, ref\_in, ref\_out, ori\_type\_out)

### 기능

‘ref\_in’ 기준 좌표계에서 표현되는 ‘pose\_in’ Task 좌표를 ‘ref\_out’ 기준 좌표계에서 표현되는 Task 좌표로 변환하여, 출력합니다. 아래의 경우에 대한 좌표변환 계산을 지원합니다.

- (ref\_in) 월드 기준 좌표계 → (ref\_out) 월드 기준 좌표계
- (ref\_in) 월드 기준 좌표계 → (ref\_out) 베이스 기준 좌표계
- (ref\_in) 월드 기준 좌표계 → (ref\_out) 툴 기준 좌표계
- (ref\_in) 월드 기준 좌표계 → (ref\_out) 사용자 기준 좌표계
- (ref\_in) 베이스 기준 좌표계 → (ref\_out) 월드 기준 좌표계
- (ref\_in) 베이스 기준 좌표계 → (ref\_out) 베이스 기준 좌표계
- (ref\_in) 베이스 기준 좌표계 → (ref\_out) 툴 기준 좌표계
- (ref\_in) 베이스 기준 좌표계 → (ref\_out) 사용자 기준 좌표계
- (ref\_in) 툴 기준 좌표계 → (ref\_out) 월드 기준 좌표계
- (ref\_in) 툴 기준 좌표계 → (ref\_out) 베이스 기준 좌표계
- (ref\_in) 툴 기준 좌표계 → (ref\_out) 툴 기준 좌표계
- (ref\_in) 툴 기준 좌표계 → (ref\_out) 사용자 기준 좌표계
- (ref\_in) 사용자 기준 좌표계 → (ref\_out) 월드 기준 좌표계
- (ref\_in) 사용자 기준 좌표계 → (ref\_out) 베이스 기준 좌표계
- (ref\_in) 사용자 기준 좌표계 → (ref\_out) 툴 기준 좌표계
- (ref\_in) 사용자 기준 좌표계 → (ref\_out) 사용자 기준 좌표계

<sup>25</sup> [https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/5014137/check\\_orientation\\_condition+axis+min+max+ref+mod+pos](https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/5014137/check_orientation_condition+axis+min+max+ref+mod+pos)

## 인수

인수명	자료형	기본값	설명
pose_in	posx	-	posx
ref_in	float	DR_COND_NONE	<p>변환 전 reference coordinate</p> <ul style="list-style-type: none"> <li>• DR_BASE : base coordinate</li> <li>• DR_WORLD : world coordinate</li> <li>• DR_TOOL : tool coordinate</li> <li>• user coordinate: 사용자 정의</li> </ul>
ref_out	float	DR_COND_NONE	<p>변환 후 reference coordinate</p> <ul style="list-style-type: none"> <li>• DR_BASE : base coordinate</li> <li>• DR_WORLD : world coordinate</li> <li>• DR_TOOL : tool coordinate</li> <li>• user coordinate: 사용자 정의</li> </ul>
ori_type_out	int	None	<p>orientation type</p> <ul style="list-style-type: none"> <li>• None: pose_in의 orientation type을 따릅니다.</li> <li>• DR_ELR_ZYZ: Euler Angles(z-y'-z'', in degrees)</li> <li>• DR_ELR_ZYX: Euler Angles(z-y'-x'', in degrees)</li> <li>• DR_ELR_XYZ: Euler Angles(x-y'-z'', in degrees)</li> <li>• DR_FIX_XYZ: Fixed Angles(x-y-z, in degrees)</li> <li>• DR_ROTVEC: 3D rotation vector (angle/axis representation)</li> <li>• DR_QUAT: unit quaternion(x, y, z, w)</li> </ul>

## 리턴

값	설명
pos	posx or pose

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시

예외	설명
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 base_pos = posx(400,500,800,0,180,15)
2 # 베이스 기준의 Task 좌표가 base_pos = posx(400,500,800,0,180,15) 인 경우
3
4 tool_pos = coord_transform(base_pos, DR_BASE, DR_TOOL)
5 # 베이스 기준의 Task 좌표인 base_pos를 툴 기준의 Task 좌표로 변환
6 # 상기 명령어는 변환된 툴 기준의 Task 좌표를 리턴 하며 tool_pos에 저장
7
8 tool_pos_zyx = coord_transform(base_pos, DR_BASE, DR_TOOL, DR_ELR_ZYX)
9 # Orientation type of tool_pos_zyx: euler zyx

```

## 관련 명령어

- [set\\_user\\_cart\\_coord\(pos, ref\)\(p. 223\)](#)
- [set\\_user\\_cart\\_coord\(u1, v1, pos, ref\)\(p. 226\)](#)
- [set\\_user\\_cart\\_coord\(x1, x2, x3, pos, ref\)\(p. 224\)](#)
- [get\\_current\\_posx\(ref, ori\\_type\)\(p. 168\)](#)
- [get\\_desired\\_posx\(ref, ori\\_type\)\(p. 178\)](#)
- [set\\_ref\\_coord\(coord\)\(p. 53\)](#)

## 6.2.17 **get\_pattern\_point(pos1, pos2, pos3, pos4, index, pattern, row, column, stack, stack\_offset, point\_offset, ori\_type\_out=None)**

### 기능

주어진 4점을 이용하여 정해진 Pattern에 맞는 Index에 대한 Pallet Point를 계산하는 기능이며, 정사각형 및 직사각형 평면 Pallet만 가능합니다. Pallet의 4점을 교시 할 때 Orientation을 고정한 후 교시 하기 바랍니다.

## 인수

인수명	자료형	기본값	설명
pos1	posx	-	posx 또는 position list
	list (float[6])		
pos2	posx	-	posx 또는 position list
	list (float[6])		
pos3	posx	-	posx 또는 position list
	list (float[6])		
pos4	posx	-	posx 또는 position list
	list (float[6])		
index	int	1	0 ~ [(row X column) - 1]
pattern	Int	0	Normal Pallet -> 0: Snake, 1: Zigzag Rhombus Pallet -> 2: Snake, 3: Zigzag
row	Int	1	줄 개수 (가로 줄 수)
column	Int	1	열 개수 (세로 줄 수)
stack	int	1	Pallet 층 개수
stack_offset	float	0.0	Pallet 층 높이 (물체의 높이 고려)
point_offset	float[3]	None	교시 점 미세조정 (Translation 방향)

인수명	자료형	기본값	설명
ori_type_out	int	None	<p>output orientation type</p> <ul style="list-style-type: none"> <li>• None: pos 인자의 orientation type을 따릅니다.</li> <li>• DR_ELR_ZYZ: Euler Angles(z-y'-z'', in degrees)</li> <li>• DR_ELR_ZYX: Euler Angles(z-y'-x'', in degrees)</li> <li>• DR_ELR_XYZ: Euler Angles(x-y'-z'', in degrees)</li> <li>• DR_FIX_XYZ: Fixed Angles(x-y-z, in degrees)</li> <li>• DR_ROTVEC: 3D rotation vector (angle/axis representation)</li> <li>• DR_QUAT: unit quaternion(x, y, z, w)</li> </ul>

## 리턴

값	설명
posx or list(float[6])	task space point

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 # Example Data
2 pos1 = posx(300,0,0,0,180,0) # Teaching pose1
3 pos2 = posx(800,0,0,0,180,0) # Teaching pose2
4 pos3 = posx(0,500,0,0,180,0) # Teaching pose3
5 pos4 = posx(800,500,0,0,180,0) # Teaching pose4
6 direction = 0 # Normal Pallet -> 0: Snake, 1: Zigzag / Rhombus Pallet ->
    2: Snake, 3: Zigzag

```

```
7  row = 3
8  column = 4
9  stack = 1
10 thickness = 0
11 point_offset = [0, 0, 0] # Offset for calculated pose
12
13 # Total count
14 if direction < 2: # Normal Pallet
15     total_count = row * column * stack
16 else: # Rhombus Pallet
17     total_count = (row * column - int(row/2)) * stack
18
19 # Calculate Pallet Pose (Resulted in base coordinate)
20 for pallet_index in range(0, total_count):
21     Pallet_Pose = get_pattern_point(pos1, pos2, pos3, pos4, pallet_index,
22                                     direction, row, column, stack, thickness, point_offset)
```

## 관련 명령어

- posx(x=0, y=0, z=0, A=0, B=0, C=0, ori\_type=None, sol=None, turn=None)(p. 30)

## 7 시스템 명령어

### 7.1 IO 관련

#### 7.1.1 set\_digital\_output(index, val=None)

##### 기능

컨트롤러의 디지털 접점에서 신호를 내보내기 위한 명령문입니다. 디지털 출력 레지스터에 저장한 값을 디지털 신호로 출력합니다.

##### 인수

인수명	자료형	기본값	설명
index	int	-	컨트롤러에 장착된 I/O 접점 번호 <ul style="list-style-type: none"> <li>val 인자가 있을 경우: 1 ~ 16까지의 숫자</li> <li>val 인자가 없을 경우: 1 ~ 16, -1 ~ -16 (양수는 ON, 음수는 OFF)</li> </ul>
val	int	-	I/O value <ul style="list-style-type: none"> <li>ON: 1</li> <li>OFF: 0</li> </ul>

##### 알아두기

val 값을 생략하면, index 인자의 부호에 따라 양수는 ON, 음수는 OFF 가 됩니다.

##### 리턴

값	설명
0	성공
음수값	실패

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 set_digital_output(1, ON) #1번 접점 ON
2 set_digital_output(16, OFF) #16번 접점 OFF
3 set_digital_output(3) #3번 접점 ON (val 인자가 생략된 경우 양수 ON)
4 set_digital_output(-3) #3번 접점 OFF (val 인자가 생략된 경우 음수 OFF)

```

## 7.1.2 set\_digital\_outputs(bit\_list)

### 기능

컨트롤러의 디지털 출력 복수 개의 접점에서 신호를 보내내기 위한 명령문입니다. bit\_list에 정의된 접점들의 디지털 신호를 한 번에 출력할 수 있습니다.

### 인수

인수명	자료형	기본값	설명
bit_list	list (int)	-	복수 개 출력하고자 하는 접점 list <ul style="list-style-type: none"> <li>양수의 접점 번호는 ON 출력: 1~16</li> <li>음수의 접점 번호는 OFF 출력: -1~-16</li> </ul>

### 리턴

값	설명
0	성공

값	설명
음수값	실패

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 set_digital_outputs(bit_list=[1,2,3,4,5,6,7,8]) #1번~8번 접점 ON
2 set_digital_outputs([-1,-2,-3,-4,-5,-6,-7,-8]) #1번~8번 접점 OFF
3 set_digital_outputs([1,-2,3]) #1번접점 ON, 2번 접점 OFF, 3번 접점 ON
4 set_digital_outputs([4,-9,-12]) #4번접점 ON, 9번 접점 OFF, 12번 접점 OFF

```

### 7.1.3 set\_digital\_outputs(bit\_start, bit\_end, val)

#### 기능

컨트롤러의 디지털 출력 시작 접점(bit\_start)부터 마지막 접점(bit\_end)까지 한 번에 복수 신호를 내보내기 위한 명령문입니다.

#### 인수

인수명	자료형	기본값	설명
bit_start	int	-	출력 신호 시작 접점 번호 (1~16)
bit_end	int	-	출력 신호 끝 접점 번호 (1~16)

인수명	자료형	기본값	설명
val	int	-	출력 값

### ❶ 알아두기

- bit\_end는 bit\_start 보다 큰 값이어야 합니다.
- val은 bit\_start =LSB, bit\_end=MSB가 되는 비트 조합의 값입니다.  
Ex) bit\_start =1, bit\_end=4, val=0b1010 # 4번=ON, 3번=OFF, 2번=ON, 1번=OFF

### 리턴

값	설명
0	성공
음수값	실패

### 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

### 예제

1	#접점1=ON, 접점2=ON, 접점3=OFF, 접점4=OFF 출력
2	set_digital_outputs(bit_start=1, bit_end=4, val=0b0011) #0b는 2진수 의미

```

3 #접점3=ON, 접점4=OFF 출력
4 set_digital_outputs(bit_start=3, bit_end=4, val=0b01) #0b는 2진수 의미
5
6 #접점1 ~ 접점 8까지 모두 ON 출력
7 set_digital_outputs(1, 8, 0xff) #0x는 16진수 의미
8

```

## 7.1.4 set\_digital\_output(index, val=None, time=None, val2=None)

### 기능

컨트롤러의 디지털 접점에서 신호를 내보내기 위한 명령문입니다. 설정된 시간만큼 지정된 신호를 내보낸 뒤, 다음 신호를 내보냅니다.

### 인수

인수명	자료형	기본값	설명
index	int	-	컨트롤러에 장착된 I/O 접점 번호 <ul style="list-style-type: none"> <li>val 인자가 있을 경우: 1 ~ 16까지의 숫자</li> <li>val 인자가 없을 경우: 1 ~ 16, -1 ~ -16</li> </ul> (양수는 ON, 음수는 OFF)
val	int	-	I/O value <ul style="list-style-type: none"> <li>ON: 1</li> <li>OFF: 0</li> </ul>
time	float	-	시간 정보(0.01 ~ 3,000,000)
val2	int	-	I/O value <ul style="list-style-type: none"> <li>ON: 1</li> <li>OFF: 0</li> </ul>

### ● 알아두기

val 값을 생략하면, index 인자의 부호에 따라 양수는 ON, 음수는 OFF 가 됩니다.

### 리턴

값	설명
0	성공

값	설명
음수값	실패

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1   set_digital_output(1, ON, 2.0, OFF) #1번 접점 ON. 2초 뒤 OFF
2   set_digital_output(5, OFF, 0.5, ON) #5번 접점 ON. 0.5초 뒤 OFF

```

## 7.1.5 get\_digital\_input(index)

### 기능

컨트롤러의 디지털 입력 점접에서 신호를 불러오기 위한 명령문으로 디지털 입력 점접 값을 읽습니다.

### 인수

인수명	자료형	기본값	설명
index	int	-	1 ~ 16까지의 숫자이며, 컨트롤러에 장착된 I/O의 접점 번호

## 리턴

값	설명
1	ON
0	OFF
음수값	실패

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1   in1 = get_digital_input(1) #1번 접점 읽기
2   in8 = get_digital_input(8) #8번 접점 읽기

```

## 7.1.6 get\_digital\_inputs(bit\_list)

### 기능

컨트롤러의 디지털 입력 복수 개의 접점에서 신호를 불러오기 위한 명령문입니다. bit\_list에 정의된 접점들의 디지털 신호를 한 번에 입력할 수 있습니다.

## 인수

인수명	자료형	기본값	설명
index	list (int)	-	복수 개로 읽어들일 input 접점 list 1 ~ 16까지의 숫자이며, 컨트롤러에 장착된 I/O 접점 번호

## 리턴

값	설명
int (>=0)	한번에 읽은 복수 개의 접점 값 (bit_list의 첫번째 값=LSB, bit_list의 마지막 값=MSB 가 되는 비트 조합의 값)
음수 값	실패

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 # input 접점: 1번=OFF, 2번=OFF, 3번=ON, 4번=ON인 경우
2 res = get_digital_inputs(bit_list=[1,2,3,4])
3 #res 기대값 = 0b1100(이진수), 12(십진수), 0x0C(16진수)
4
5 # input 접점: 5번=ON, 6번=ON, 7번=OFF, 8번=ON인 경우
6 res = get_digital_inputs([5,6,7,8])
7 #res 기대값 = 0b1011(이진수), 11(십진수), 0x0B(16진수)

```

## 7.1.7 get\_digital\_inputs(bit\_start, bit\_end)

### 기능

컨트롤러의 디지털 입력 시작 점점(start\_index)부터 마지막 점점(end\_index)까지 한 번에 복수 신호를 불러오기 위한 명령문입니다.

### 인수

인수명	자료형	기본값	설명
bit_start	int	-	입력 신호 시작 점점 번호 (1~16)
bit_end	int	-	입력 신호 끝 점점 번호 (1~16)

#### 알아두기

bit\_end는 bit\_start 보다 큰 값이어야 합니다.

### 리턴

값	설명
int ( $\geq 0$ )	한번에 읽은 복수 개의 점점 값 bit_start=LSB, bit_end=MSB 가 되는 비트 조합의 값
음수 값	실패

### 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시

예외	설명
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 # input 접점: 1번=OFF, 2번=OFF, 3번=ON, 4번=ON인 경우
2 res = get_digital_inputs(bit_start=1, bit_end=4)
3 #res 기대값 = 0b1100(이진수), 12(십진수), 0x0C(16진수)

```

## 7.1.8 wait\_digital\_input(index, val, timeout=None)

### 기능

컨트롤러의 디지털 입력 레지스터의 신호값이 val(ON or OFF)이 될 때까지 대기합니다. 대기 시간은 timeout 설정으로 변경할 수 있으며, 지정된 시간이 지나면 대기 상태가 종료됨과 동시에 결과를 리턴합니다. 단, timeout을 설정하지 않으면 무한 대기합니다.

### 인수

인수명	자료형	기본값	설명
index	int	-	1 ~ 16까지의 숫자이며, 컨트롤러에 장착됨 I/O index
value	int	-	I/O value <ul style="list-style-type: none"> <li>• ON : 1</li> <li>• OFF : 0</li> </ul>
timeout	float	-	대기 시간 [sec] 설정하지 않으면 무한 대기

### 리턴

값	설명
0	성공
-1	실패 (time-out)

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 wait_digital_input(1, ON) #1번 접점이 ON될 때까지 무한 대기
2 wait_digital_input(2, OFF) #2번 접점이 OFF될 때까지 무한 대기
3 res = wait_digital_input(1, ON, 3) #1번 접점이 ON될 때까지 3초간 대기
4 #3초 안에 1번 접점이 ON 되면, 대기 중지, res = 0
5 #3초 안에 1번 접점이 ON 되지 않았으면, 대기 중지, res = -1

```

## 7.1.9 set\_tool\_digital\_output(index, val=None)

### 기능

로봇 툴의 신호를 디지털 접점에서 내보내기 위한 명령문입니다.

#### ⚠ 주의

로봇 모델이나 플랜지 보드 버전에 따라 index 파라미터의 범위가 변경될 수 있습니다.

## 인수

인수명	자료형	기본값	설명
index	int	-	로봇 암에 장착된 I/O 접점 번호 val 인자가 있을 경우: 1 ~ 6까지의 숫자 • val 인자가 없을 경우: 1 ~ 6, -1 ~ -6 (양수는 ON, 음수는 OFF)
val	int	-	I/O value : 출력하고자 하는 값

**i 알아두기**

val 값을 생략하면, index 인자의 부호에 따라 양수는 ON, 음수는 OFF 가 됩니다.

## 리턴

값	설명
0	성공
음수값	오류

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 set_tool_digital_outputs(1, ON) #로봇 암의 1번 접점 ON
2 set_tool_digital_output(6, OFF) #로봇 암의 6번 접점 OFF
3 set_tool_digital_output(3) #3번 접점 ON, val 인자가 생략된 경우 양수 ON
4 set_tool_digital_output(-3) #3번 접점 OFF, val 인자가 생략된 경우 음수 OFF

```

### 7.1.10 set\_tool\_digital\_outputs(bit\_list)

#### 기능

로봇 툴의 신호를 디지털 접점에서 내보내기 위한 명령문으로 bit\_list에 정의된 접점들의 디지털 신호를 한 번에 출력할 수 있습니다.

#### 인수

인수명	자료형	기본값	설명
bit_list	list (int)	-	복수 개를 출력하고자 하는 접점 list <ul style="list-style-type: none"> <li>양수의 접점 번호는 ON 출력: 1~6</li> <li>음수의 접점 번호는 OFF 출력: -1~-6</li> </ul>

#### 리턴

값	설명
0	성공
음수값	오류

#### 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시

예외	설명
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 set_tool_digital_outputs(bit_list=[1,2,3,4,5,6]) #1번~6번 접점 ON
2 set_tool_digital_outputs([-1,-2,-3,-4,-5,-6]) #1번~6번 접점 OFF
3 set_tool_digital_outputs([1,-2,3]) #1번접점 ON, 2번 접점 OFF, 3번 접점 ON

```

### 7.1.11 set\_tool\_digital\_outputs(bit\_start, bit\_end, val)

#### 기능

로봇 툴의 신호를 디지털 접점에서 내보내기 위한 명령문으로 시작 접점(bit\_start)부터 마지막 접점(bit\_end)까지 복수 신호를 한 번에 출력할 수 있습니다.

#### 인수

인수명	자료형	기본값	설명
bit_start	int	-	출력 신호 시작 접점 번호 (1~6)
bit_end	int	-	출력 신호 끝 접점 번호 (1~6)
Val	int	-	출력 값

#### ● 알아두기

- bit\_end는 bit\_start 보다 큰 값이어야 합니다.
- val은 bit\_start =LSB, bit\_end=MSB가 되는 비트 조합의 값입니다.  
Ex) bit\_start =1, bit\_end=4, val=0b1010 # 4번=ON, 3번=OFF, 2번=ON, 1번=OFF

## 리턴

값	설명
0	성공
음수값	오류

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 #접점1=ON, 접점2=ON, 접점3=OFF, 접점4=OFF 출력
2 set_tool_digital_outputs(bit_start=1, bit_end=4, val=0b0011) #0b는 2진수를 의미
3
4 #접점3=ON, 접점4=OFF 출력
5 set_tool_digital_outputs(bit_start=3, bit_end=4, val=0b01) #0b는 2진수를 의미
6
7 #접점1 ~ 접점 8까지 모두 ON 출력
8 set_tool_digital_outputs(1, 8, 0xff) #0x는 16진수를 의미

```

## 7.1.12 set\_tool\_digital\_output(index, val=None, time=None, val2=None)

### 기능

로봇 툴의 신호를 디지털 접점에서 내보내기 위한 명령문입니다. 설정된 시간만큼 지정된 신호를 내보낸 뒤, 다음 신호를 내보냅니다.

### 주의

로봇 모델이나 플랜지 보드 버전에 따라 index 파라미터의 범위가 변경될 수 있습니다.

## 인수

인수명	자료형	기본값	설명
index	int	-	컨트롤러에 장착된 I/O 접점 번호 <ul style="list-style-type: none"> <li>val 인자가 있을 경우: 1 ~ 16까지의 숫자</li> <li>val 인자가 없을 경우: 1 ~ 16, -1 ~ -16</li> </ul> (양수는 ON, 음수는 OFF)
val	int	-	I/O value <ul style="list-style-type: none"> <li>ON: 1</li> <li>OFF: 0</li> </ul>
time	float	-	시간 정보(0.01 ~ 3,000,000)
val2	int	-	I/O value <ul style="list-style-type: none"> <li>ON: 1</li> <li>OFF: 0</li> </ul>

### 알아두기

val 값을 생략하면, index 인자의 부호에 따라 양수는 ON, 음수는 OFF 가 됩니다.

## 리턴

값	설명
0	성공
음수값	실패

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시

예외	설명
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 set_tool_digital_outputs(1, ON, 2.0, OFF) #로봇 암의 1번 접점 ON. 2초 뒤 OFF
2 set_tool_digital_output(5, OFF, 0.5, ON) #로봇 암의 5번 접점 OFF. 0.5초 뒤 ON

```

### 7.1.13 get\_tool\_digital\_input(index)

#### 기능

로봇 툴의 신호를 디지털 접점에서 불러오기 위한 명령문입니다.

#### 인수

인수명	자료형	기본값	설명
index	int	-	로봇 Tool I/O 접점 번호 (1~6)

#### 리턴

값	설명
1	ON
0	OFF
음수값	실패

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1  get_tool_digital_input(1) #Tool IO 입력 1번 접점 읽기
2  get_tool_digital_input(6) #Tool IO 입력 6번 접점 읽기

```

### 7.1.14 get\_tool\_digital\_inputs(bit\_list)

#### 기능

로봇 툴의 신호를 디지털 입력 접점에서 불러오기 위한 명령문으로 bit\_list에 정의된 접점들의 디지털 신호를 한 번에 입력할 수 있습니다.

#### 인수

인수명	자료형	기본값	설명
bit_list	list (int)	-	복수 개로 읽어들일 input 접점 list • (로봇 암에 장착된 I/O 입력 접점 번호, 1~6)

## 리턴

값	설명
int (>=0)	한 번에 읽은 복수 개의 접점 값 (bit_list 의 첫번째 값=LSB, bit_list 의 마지막 값=MSB 가 되는 비트 조합의 값)
음수 값	실패

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 # input 접점: 1번=OFF, 2번=OFF, 3번=ON인 경우
2 res = get_tool_digital_inputs(bit_list=[1,2,3]) # 1,2,3번 접점 한 번에 읽기
3 #res 기대값 = 0b100(이진수), 4(십진수), 0x04(16진수)
4
5 # input 접점: 4번=ON, 5번=ON, 6번=OFF인 경우
6 res = get_tool_digital_inputs([4,5,6])
7 #res 기대값 = 0b011(이진수), 3(십진수), 0x03(16진수)

```

### 7.1.15 get\_tool\_digital\_inputs(bit\_start, bit\_end)

## 기능

로봇 툴의 신호를 디지털 접점에서 불러오기 위한 명령문으로 시작 접점(start\_index)부터 마지막 접점(end\_index) 까지 복수 신호를 한 번에 입력할 수 있습니다.

## 인수

인수명	자료형	기본값	설명
bit_start	int	-	입력 신호 시작 접점 번호 (1~6)
bit_end	int	-	입력 신호 끝 접점 번호 (1~6)

## 리턴

값	설명
int (>=0)	한번에 읽은 복수 개의 접점 값 bit_start=LSB, bit_end=MSB 가 되는 비트 조합의 값
음수 값	실패

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 # input 접점: 1번=OFF, 2번=OFF, 3번=ON인 경우
2 res = get_tool_digital_inputs(bit_start=1, bit_end=3)
3 # res 기대값 = 0b100(이진수), 4(십진수), 0x04(16진수)
4
5 # input 접점: 4번=ON, 5번=ON, 6번=OFF인 경우
6 res = get_tool_digital_inputs(4, 6)

```

7	#res 기대값 = 0b011(이진수), 3(십진수), 0x03(16진수)
---	---

### 7.1.16 wait\_tool\_digital\_input(index, val, timeout=None)

#### 기능

로봇 툴의 디지털 입력 신호값이 val(ON or OFF)이 될 때까지 대기합니다. 대기 시간은 timeout 설정으로 변경할 수 있으며, 지정된 시간이 지나면 대기 상태가 종료됨과 동시에 결과를 리턴합니다. 단, timeout을 설정하지 않으면 무한 대기합니다.

#### 인수

인수명	자료형	기본값	설명
index	int	-	1 ~ 6까지의 숫자이며, 로봇 암에 장착됨 I/O index
value	int	-	I/O value ↳ ON : 1 ↳ OFF : 0
timeout	float	-	대기 시간 [sec] 설정하지 않으면 무한 대기

#### 리턴

값	설명
0	성공
-1	실패 (time-out)

#### 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시

예외	설명
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1  wait_tool_digital_input(1, ON) #1번 접점이 ON될 때까지 무한 대기
2  wait_tool_digital_input(2, OFF) #2번 접점이 OFF될 때까지 무한 대기
3
4  res = wait_tool_digital_input(1, ON, 3) #1번 접점이 ON될 때까지 3초간 대기
5  #3초 안에 1번 접점이 ON 되면, 대기 중지, res = 0
6  #3초 안에 1번 접점이 ON 되지 않았으면 대기 중지, res = -1

```

## 7.1.17 set\_mode\_analog\_output(ch, mod)

### 기능

컨트롤러 아날로그 출력에 대한 채널 모드를 설정합니다.

### 인수

인수명	자료형	기본값	설명
ch	int	-	<ul style="list-style-type: none"> <li>• 1 : channel 1</li> <li>• 2 : channel 2</li> </ul>
mod	int	-	analog io mode <ul style="list-style-type: none"> <li>• DR_ANALOG_CURRENT: 전류 모드</li> <li>• DR_ANALOG_VOLTAGE: 전압 모드</li> </ul>

### 리턴

값	설명
0	성공
음수값	실패

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 # analog_output channel 1을 전류 모드로 설정함
2 set_mode_analog_output(ch=1, mod=DR_ANALOG_CURRENT)
3
4 # analog_output channel 2를 전압 모드로 설정함
5 set_mode_analog_output(ch=2, mod=DR_ANALOG_VOLTAGE)

```

### 7.1.18 set\_mode\_analog\_input(ch, mod )

#### 기능

컨트롤러 아날로그 입력 대한 채널 모드를 설정합니다.

#### 인수

인수명	자료형	기본값	설명
ch	int	-	<ul style="list-style-type: none"> <li>• 1 : channel 1</li> <li>• 2 : channel 2</li> </ul>
mod	int	-	analog io mode <ul style="list-style-type: none"> <li>• DR_ANALOG_CURRENT: 전류 모드</li> <li>• DR_ANALOG_VOLTAGE: 전압 모드</li> </ul>

## 리턴

값	설명
0	성공
음수값	실패

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 # analog input channel 1을 전류 모드로 설정함
2 set_mode_analog_input(ch=1, mod=DR_ANALOG_CURRENT)
3
4 # analog input channel 2를 전압 모드로 설정함.
5 set_mode_analog_input(ch=2, mod=DR_ANALOG_VOLTAGE)

```

## 7.1.19 set\_analog\_output(ch, val)

### 기능

컨트롤러 아날로그 출력에 해당하는 채널의 값을 출력합니다.

## 인수

인수명	자료형	기본값	설명
ch	int	-	<ul style="list-style-type: none"> <li>• 1 : channel 1</li> <li>• 2 : channel 2</li> </ul>
val	float	-	analog 출력 값 <ul style="list-style-type: none"> <li>• 전류 모드인 경우: 4.0~20.0 [mA]</li> <li>• 전압 모드인 경우: 0~10.0 [V]</li> </ul>

## 리턴

값	설명
0	성공
음수값	실패

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 set_mode_analog_output(ch=1, mod=DR_ANALOG_CURRENT) #out ch1=current mode
2 set_mode_analog_output(ch=2, mod=DR_ANALOG_VOLTAGE) #out ch1=voltage mode
3
4 set_analog_output(ch=1, val=5.2) #channel 1에 5.2 mA 출력

```

5	set_analog_output(ch=2, val=10.0) #channel 2에 10V 출력
---	--

## 7.1.20 get\_analog\_input(ch)

### 기능

컨트롤러 아날로그 입력에 해당하는 채널의 값을 불러옵니다.

### 인수

인수명	자료형	기본값	설명
ch	int	-	<ul style="list-style-type: none"> <li>• 1 : channel 1</li> <li>• 2 : channel 2</li> </ul>

### 리턴

값	설명
float	해당 channel의 analog input 값 <ul style="list-style-type: none"> <li>• 전류 모드인 경우: 4.0~20.0 [mA]</li> <li>• 전압 모드인 경우: 0~10.0 [V]</li> </ul>

### 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 set_mode_analog_input(ch=1, mod=DR_ANALOG_CURRENT) #input ch1=current mode
2 set_mode_analog_input(ch=2, mod=DR_ANALOG_VOLTAGE) #input ch2=voltage mode
3
4 Cur = get_analog_input(1) # channel 1의 analog input 전류 값 읽기
5 Vol = get_analog_input(2) # channel 2의 analog input 전압 값 읽기

```

### 7.1.21 set\_output(port\_type, index, val=None, time=None, val2=None)

#### 기능

컨트롤러/플랜지의 디지털/아날로그 접점에서 신호를 출력하기 위한 명령어입니다.

#### 주의

현재 플랜지 보드 버전 및 로봇 모델에 따라 파라미터가 변경될 수 있습니다. 매뉴얼 숙지 바랍니다.

#### 인수

인수명	자료형	기본값	설명
port_type	int	-	<p>명령어를 적용할 포트 타입</p> <ul style="list-style-type: none"> <li>• DR_CONTROLLER_DIGITAL : 0</li> <li>• DR_FLANGE_DIGITAL : 1</li> <li>• DR_CONTROLLER_ANALOG : 2</li> </ul>
index	int	-	<p>출력할 I/O의 번호(Analog 출력을 위한 Channel 역할도 수행)</p> <ul style="list-style-type: none"> <li>• port_type이 DR_CONTROLLER_DIGITAL일 경우           <ul style="list-style-type: none"> <li>• 1 ~ 16</li> <li>• -16 ~ -1(val이 None일 경우) : 양수는 ON / 음수는 OFF</li> </ul> </li> <li>• port_type이 DR_FLANGE_DIGITAL일 경우           <ul style="list-style-type: none"> <li>• 구 플랜지 M/H 모델의 경우               <ul style="list-style-type: none"> <li>• 1 ~ 6</li> <li>• -6 ~ -1(val이 None일 경우) : 양수는 ON / 음수는 OFF</li> </ul> </li> <li>• 구 플랜지 / 신규 플랜지 A 모델일 경우               <ul style="list-style-type: none"> <li>• 1 ~ 2</li> <li>• -2 ~ -1(val이 None일 경우) : 양수는 ON / 음수는 OFF</li> </ul> </li> </ul> </li> <li>• 신규 플랜지 M/H 모델의 경우           <ul style="list-style-type: none"> <li>• 1 ~ 4</li> <li>• -4 ~ -1(val이 None일 경우) : 양수는 ON / 음수는 OFF</li> </ul> </li> <li>• port_type이 DR_CONTROLLER_ANALOG 일 경우           <ul style="list-style-type: none"> <li>• 1 ~ 2(channel)</li> </ul> </li> </ul>

인수명	자료형	기본값	설명
val	float	None	I/O value <ul style="list-style-type: none"> <li>port_type이 0, 1(digital)일 경우 <ul style="list-style-type: none"> <li>ON : 1</li> <li>OFF : 0</li> </ul> </li> <li>port_type이 2(analog)일 경우 <ul style="list-style-type: none"> <li>전류 모드인 경우 : 4.00 ~ 20.00(mA)</li> <li>전압 모드인 경우 : 0.00 ~ 10.00(V)</li> </ul> </li> </ul>
time	float	None	시간 정보(port_type이 0, 1일 경우에만 사용 가능) <ul style="list-style-type: none"> <li>0.01 ~ 3,000,000(sec)</li> </ul>
val2	int	None	I/O value <ul style="list-style-type: none"> <li>ON: 1</li> <li>OFF: 0</li> </ul>

### ❶ 알아두기

val 값을 생략하면, index 인자의 부호에 따라 양수는 ON, 음수는 OFF 가 됩니다.

### 리턴

값	설명
0	성공
음수값	실패

### 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 set_output(1, -1) ## set_tool_digital_output(-1)과 동일
2 set_output(0, 2, OFF, 3, ON) ## set_digital_output(2, OFF, 3, ON)과 동일
3 set_mode_analog_output(ch=2, mod=DR_ANALOG_VOLTAGE) #out ch1=voltage mode
4 set_output(2, 1, 10.0) ## set_analog_output(2, 10)와 동일

```

### 7.1.22 get\_input(port\_type, index)

#### 기능

컨트롤러/플랜지의 디지털/아날로그 접점에서 수신 된 신호를 확인하기 위한 명령어입니다.

#### 주의

현재 플랜지 보드 버전 및 로봇 모델에 따라 파라미터가 변경될 수 있습니다. 매뉴얼 숙지 바랍니다.

#### 인수

인수명	자료형	기본값	설명
port_type	int	-	<p>명령어를 적용할 포트 타입</p> <ul style="list-style-type: none"> <li>• DR_CONTROLLER_DIGITAL : 0</li> <li>• DR_FLANGE_DIGITAL : 1</li> <li>• DR_CONTROLLER_ANALOG : 2</li> <li>• DR_FLANGE_ANALOG : 3</li> </ul>

인수명	자료형	기본값	설명
index	int	-	<p>출력할 I/O의 번호(Analog 출력을 위한 Channel 역할도 수행)</p> <ul style="list-style-type: none"> <li>port_type이 DR_CONTROLLER_DIGITAL일 경우           <ul style="list-style-type: none"> <li>1 ~ 16</li> <li>-16 ~ -1(val이 None일 경우) : 양수는 ON / 음수는 OFF</li> </ul> </li> <li>port_type이 DR_FLANGE_DIGITAL일 경우           <ul style="list-style-type: none"> <li>구 플랜지 M/H 모델의 경우               <ul style="list-style-type: none"> <li>1 ~ 6</li> <li>-6 ~ -1(val이 None일 경우) : 양수는 ON / 음수는 OFF</li> </ul> </li> <li>구 플랜지 / 신규 플랜지 A 모델일 경우               <ul style="list-style-type: none"> <li>1 ~ 2</li> <li>-2 ~ -1(val이 None일 경우) : 양수는 ON / 음수는 OFF</li> </ul> </li> <li>신규 플랜지 M/H 모델의 경우               <ul style="list-style-type: none"> <li>1 ~ 4</li> <li>-4 ~ -1(val이 None일 경우) : 양수는 ON / 음수는 OFF</li> </ul> </li> </ul> </li> <li>port_type이 DR_CONTROLLER_ANALOG 일 경우           <ul style="list-style-type: none"> <li>1 ~ 2(channel)</li> </ul> </li> <li>port_type이 DR_FLANGE_ANALOG일 경우(신규 플랜지에서만 사용 가능)           <ul style="list-style-type: none"> <li>1 ~ 2(A 모델일 경우)</li> <li>1 ~ 4(M/H 모델일 경우)</li> </ul> </li> </ul>

## 리턴

값	설명
1	(디지털) ON
0	(디지털) OFF
float	(아날로그) 전류 모드인 경우 4.0 ~ 20.0(mA) 전압 모드인 경우 0.0 ~ 10.0(V)
음수값	실패

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 res1 = get_input(0, 1) ## get_digital_input(1)과 동일
2 res2 = get_input(2, 1) ## get_analog_input(1)과 동일
3 res3 = get_input(3, 1) ## get_tool_analog_input(1)과 동일

```

### 7.1.23 wait\_input(port\_type, index, val, timeout=None, condition=None)

#### 기능

컨트롤러/플랜지의 디지털/아날로그 접점에서 수신 된 신호값이 val이 될 때까지 대기합니다. 대기 시간은 timeout 설정으로 변경할 수 있으며, 지정된 시간이 지나면 대기 상태가 종료됨과 동시에 결과를 리턴합니다. 단, timeout을 설정하지 않으면 무한 대기합니다.

#### ⚠ 주의

현재 플랜지 보드 버전 및 로봇 모델에 따라 파라미터가 변경될 수 있습니다. 매뉴얼 숙지 바랍니다.

#### 인수

인수명	자료형	기본값	설명
port_type	int	-	<p>명령어를 적용할 포트 타입</p> <ul style="list-style-type: none"> <li>• DR_CONTROLLER_DIGITAL : 0</li> <li>• DR_FLANGE_DIGITAL : 1</li> <li>• DR_CONTROLLER_ANALOG : 2</li> <li>• DR_FLANGE_ANALOG : 3</li> </ul>

인수명	자료형	기본값	설명
index	int	-	<p>출력할 I/O의 번호(Analog 출력을 위한 Channel 역할도 수행)</p> <ul style="list-style-type: none"> <li>port_type이 DR_CONTROLLER_DIGITAL일 경우           <ul style="list-style-type: none"> <li>1 ~ 16</li> <li>-16 ~ -1(val이 None일 경우) : 양수는 ON / 음수는 OFF</li> </ul> </li> <li>port_type이 DR_FLANGE_DIGITAL일 경우           <ul style="list-style-type: none"> <li>구 플랜지 M/H 모델의 경우               <ul style="list-style-type: none"> <li>1 ~ 6</li> <li>-6 ~ -1(val이 None일 경우) : 양수는 ON / 음수는 OFF</li> </ul> </li> <li>구 플랜지 / 신규 플랜지 A 모델일 경우               <ul style="list-style-type: none"> <li>1 ~ 2</li> <li>-2 ~ -1(val이 None일 경우) : 양수는 ON / 음수는 OFF</li> </ul> </li> <li>신규 플랜지 M/H 모델의 경우               <ul style="list-style-type: none"> <li>1 ~ 4</li> <li>-4 ~ -1(val이 None일 경우) : 양수는 ON / 음수는 OFF</li> </ul> </li> </ul> </li> <li>port_type이 DR_CONTROLLER_ANALOG 일 경우           <ul style="list-style-type: none"> <li>1 ~ 2(channel)</li> </ul> </li> <li>port_type이 DR_FLANGE_ANALOG일 경우(신규 플랜지에서만 사용 가능)           <ul style="list-style-type: none"> <li>1 ~ 2(A 모델일 경우)</li> <li>1 ~ 4(M/H 모델일 경우)</li> </ul> </li> </ul>
val	float	-	I/O value
timeout	float	None	<p>대기 시간 [sec]</p> <p>설정하지 않으면 무한 대기</p>
condition	int	None	<p>Analog 값 비교 조건</p> <ul style="list-style-type: none"> <li>DR_ANALOG_CONDITION_UPPER : 0           <ul style="list-style-type: none"> <li>수신된 analog input이 val 파라미터보다 크거나 같아질 때 까지 대기</li> </ul> </li> <li>DR_ANALOG_CONDITION_LOWER : 1           <ul style="list-style-type: none"> <li>수신된 analog input이 val 파라미터보다 작거나 같아질 때 까지 대기</li> </ul> </li> </ul>

## 리턴

값	설명
0	성공
-1	실패 (time-out)

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 wait_input(0, 1, ON) # wait_digital_input(1, ON)과 동일
2 wait_input(1, 2, ON) # wait_tool_digital_input(2, ON)과 동일
3 wait_input(2, 1, 4.0, DR_ANALOG_CONDITION_LOWER) # wait_analog_input(1,
DR_ANALOG_CONDITION_LOWER, 4.0)과 동일

```

### 7.1.24 wait\_analog\_input(ch, condition, val, timeout=None)

#### 기능

컨트롤러의 아날로그 입력 채널의 신호값이 val(float 값)이 될 때까지 대기합니다. 대기 시간은 timeout 설정으로 변경할 수 있으며, 지정된 시간이 지나면 대기 상태가 종료됨과 동시에 결과를 리턴합니다. 단, timeout을 설정하지 않으면 무한 대기합니다.

#### 인수

인수명	자료형	기본값	설명
ch	int	-	1 : channel 1 2 : channel 2
condition	int	None	Analog 값 비교 조건 <ul style="list-style-type: none"> <li>• DR_ANALOG_CONDITION_UPPER : 0 <ul style="list-style-type: none"> <li>• 수신된 analog input0  val 파라미터보다 크거나 같아 질 때 까지 대기</li> </ul> </li> <li>• DR_ANALOG_CONDITION_LOWER : 1 <ul style="list-style-type: none"> <li>• 수신된 analog input0  val 파라미터보다 작거나 같아 질 때 까지 대기</li> </ul> </li> </ul>

인수명	자료형	기본값	설명
val	float	-	수신된 analog input 데이터와 비교할 비교값
timeout	float	None	대기 시간 [sec] 설정하지 않으면 무한 대기

## 리턴

값	설명
0	성공
-1	실패 (time-out)

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

1	wait_analog_input(1, DR_ANALOG_CONDITION_UPPER, 5.0, 4) ## ch 1번으로 수신된 analog 데이터가 5.0과 같거나 커질 때 까지 대기
---	---

## 7.1.25 wait\_tool\_analog\_input(ch, condition, val, timeout=None)

### 기능

플랜지 아날로그 입력 채널의 신호값이 val(float 값)이 될 때까지 대기합니다. 대기 시간은 timeout 설정으로 변경할 수 있으며, 지정된 시간이 지나면 대기 상태가 종료됨과 동시에 결과를 리턴합니다. 단, timeout을 설정하지 않으면 무한 대기합니다.

**⚠ 주의**

해당 명령어는 신규 플랜지에서만 사용 가능합니다.

## 인수

인수명	자료형	기본값	설명
ch	int	-	1 : channel 1 2 : channel 2 3 : chaanel 3 (M/H 모델만 가능) 4 : channel 4 (M/H 모델만 가능)
condition	int	None	Analog 값 비교 조건 <ul style="list-style-type: none"> <li>• DR_ANALOG_CONDITION_UPPER : 0<ul style="list-style-type: none"> <li>• 수신된 analog input0  val 파라미터보다 크거나 같아 질 때 까지 대기</li> </ul></li> <li>• DR_ANALOG_CONDITION_LOWER : 1<ul style="list-style-type: none"> <li>• 수신된 analog input0  val 파라미터보다 작거나 같아 질 때 까지 대기</li> </ul></li> </ul>
val	float	-	수신된 analog input 데이터와 비교할 비교값
timeout	float	None	대기 시간 [sec] 설정하지 않으면 무한 대기

## 리턴

값	설명
0	성공
-1	실패 (time-out)

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시

예외	설명
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```
1    wait_tool_analog_input(1, DR_ANALOG_CONDITION_UPPER, 5.0, 4) ## ch 1번으로
수신된 analog 데이터가 5.0과 같거나 커질 때 까지 대기
```

## 7.2 TP 연동

### 7.2.1 tp\_popup(message, pm\_type=DR\_PM\_MESSAGE, button\_type=0)

#### 기능

터치펜던트를 통해 사용자에게 메시지를 제공합니다. 상위 제어기는 String을 받아서 팝업 창에 표시하며, 사용자의 확인 작업에 의해 창이 닫혀야 합니다.

#### 인수

인수명	자료형	기본값	설명
message	string	-	<p>사용자에게 제공할 메시지</p> <ul style="list-style-type: none"> <li>메시지는 256byte 이내로 제한됩니다.</li> <li>텍스트는 간략하게 작성하시길 권장합니다. 긴 텍스트의 경우 일부 내용이 줄임표(...)로 생략됩니다.</li> <li>새줄(\n) 또는 캐리지 리턴(\r) 등의 포맷팅 관련 코드는 사용할 수 없습니다.</li> </ul>
pm_type	int	DR_PM_MESSAGE	<p>메시지 유형</p> <ul style="list-style-type: none"> <li>DR_PM_MESSAGE</li> <li>DR_PM_WARNING</li> <li>DR_PM_ALARM</li> </ul>
button_type	int	0	<p>TP pop 메시지 버튼 type</p> <ul style="list-style-type: none"> <li>0 : Stop &amp; Resume 버튼 표시</li> <li>1 : Stop 버튼 표시</li> </ul>

## 리턴

값	설명
0	성공
음수값	실패

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 tp_popup("move done", DR_PM_MESSAGE)
2 tp_popup("Error!! ", DR_PM_ALARM)
3 a=1
4 b=2
5 c=3
6 tp_popup("a={0}, b={1}, c={2}!".format(a,b,c) ,DR_PM_MESSAGE)
7 tp_popup("critical error!! ", DR_PM_ALARM, 1)

```

## 7.2.2 tp\_log(message)

### 기능

티치펜던트에 사용자가 작성한 로그를 기록합니다.

## 인수

인수명	자료형	기본값	설명
message	string	-	<p>로그 메시지</p> <ul style="list-style-type: none"> <li>메시지는 256byte 이내로 제한됩니다.</li> <li>텍스트는 간략하게 작성하시길 권장합니다. 긴 텍스트의 경우 일부 내용이 줄임표(...)로 생략됩니다.</li> <li>새줄(\n) 또는 캐리지 리턴(\r) 등의 포맷팅 관련 코드는 사용할 수 없습니다.</li> </ul>

## 리턴

값	설명
0	성공
음수값	실패

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```
1 tp_log("movej() is complete! ")
```

### 7.2.3 tp\_get\_user\_input(message, input\_type)

## 기능

터치펜던트를 통해 사용자 입력 정보를 받습니다.

## 인수

인수명	자료형	기본값	설명
message	string	-	TP 사용자 입력창에 표시될 문자열 메시지 <ul style="list-style-type: none"> <li>메시지는 256byte 이내로 제한됩니다.</li> <li>텍스트는 간략하게 작성하시길 권장합니다. 긴 텍스트의 경우 일부 내용이 줄임표 (...)로 생략됩니다.</li> <li>새줄(\n) 또는 캐리지 리턴(\r) 등의 포맷팅 관련 코드는 사용할 수 없습니다.</li> </ul>
input_type	int	-	TP 사용자 입력 메시지 타입 <ul style="list-style-type: none"> <li>DR_VAR_INT: 정수형</li> <li>DR_VAR_FLOAT: 실수형</li> <li>DR_VAR_STR: 문자열</li> <li>DR_VAR_BOOL: 불리언</li> </ul>

## 리턴

값	설명
사용자 입력 데이터	TP로 받은 사용자 입력 데이터

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 q1 = posj(10, 10, 10, 10, 10, 10)
2 q2 = posj(20, 20, 20, 20, 20, 20)
3 q3 = posj(30, 30, 30, 30, 30, 30)
4 q4 = posj(40, 40, 40, 40, 40, 40)

```

```

5   q5 = posj(50, 50, 50, 50, 50, 50)
6   q6 = posj(60, 60, 60, 60, 60, 60)
7
8   int_y= tp_get_user_input("message1", input_type= DR_VAR_INT)
9   if int_y==1: #TP 사용자 입력이 1로 들어온 경우 q1으로 이동
10  movej(q1, vel=30, acc=30)
11  else: #TP 사용자 입력이 1로 들어오지 않은 경우 q2으로 이동
12  movej(q2, vel=30, acc=30)
13
14  float_y= tp_get_user_input("message2", input_type= DR_VAR_FLOAT)
15  if float_y==3.14: #TP 사용자 입력이 3.14로 들어온 경우 q3으로 이동
16  movej(q3, vel=30, acc=30)
17  else: #TP 사용자 입력이 3.14로 들어오지 않은 경우 q4으로 이동
18  movej(q4, vel=30, acc=30)
19
20  str_y= tp_get_user_input("message3", input_type= DR_VAR_STR)
21  if str_y=="a": #TP 사용자 입력이 "a"로 들어온 경우 q5으로 이동
22  movej(q5, vel=30, acc=30)
23  else: #TP 사용자 입력이 "a"로 들어오지 않은 경우 q6으로 이동
24  movej(q6, vel=30, acc=30)
25
26  bool_y= tp_get_user_input("message3", input_type= DR_VAR_BOOL)
27  if bool_y==True: #TP 사용자 입력이 True or 1로 들어온 경우 q5으로 이동
28  movej(q5, vel=30, acc=30)
29  else: #TP 사용자 입력이 False or 0로 들어온 경우 q6으로 이동
30  movej(q6, vel=30, acc=30)

```

## 7.3 Thread

### 7.3.1 thread\_run(th\_func\_name, loop=False)

#### 기능

Thread를 생성하여 수행하며 Thread가 수행할 기능은 th\_func\_name에 지정된 함수에 따라 결정됩니다.

#### 알아두기

Thread 명령어를 사용할 때, 제약 조건은 아래와 같습니다.

- Thread는 최대 4개까지만 사용 가능합니다.
- Thread 내에서 로봇을 움직이는 하기 모션 명령어는 사용할 수 없습니다.
  - movej, amovej, movejx, amovejx, movel, amovel, movec, amovec, movesj, amovesj,
  - movesx, amovesx, moveb, amoveb, move\_spiral, amove\_spiral,
  - move\_periodic, amove\_periodic

- thread\_run 시 loop=True 하고, 해당 쓰레드 함수 내에서 무한루프로 블록되어 있는 경우에는 쓰레드 명령들이 정상 동작하지 않습니다. (단, TP를 통한 STOP시, 해당 쓰레드는 정상적으로 종료됩니다.)

## 인수

인수명	자료형	기본값	설명
th_func_name	callable	-	Thread가 수행할 function name
loop	bool	False	Thread의 반복 수행 여부 <ul style="list-style-type: none"> <li>• True : th_func_name이 반복적 호출 (interval 0.01second)</li> <li>• False : th_func_name이 1회 호출</li> </ul>

## 리턴

값	설명
int	등록된 thread ID
음수값	실패

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1  ----- 쓰레드 -----
2  def fn_th_func():
3      if check_motion() == 0: # 수행 중인 모션이 없는 경우
4          set_digital_output(1, OFF)
5      else:

```

```

6   set_digital_output(1, ON)
7
8   ----- 메인 루틴 -----
9   th_id = thread_run(fn_th_func, loop=True) #쓰레드 Run
10
11  while 1:
12      # do something...
13      wait(0.1)

```

### 7.3.2 thread\_stop(th\_id)

#### 기능

Thread를 종료합니다.

DRL 프로그램이 종료될 경우에 thread\_stop() 명령을 사용하지 않아도 자동으로 종료됩니다.

#### 인수

인수명	자료형	기본값	설명
th_id	int	-	Stop 하고자 하는 thread ID

#### 리턴

값	설명
0	성공
음수값	실패

#### 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 def fn_th_func():
2     if check_motion() == 0: # 수행 중인 모션이 없는 경우
3         set_digital_output(1, OFF)
4     else:
5         set_digital_output(1, ON)
6         ----- 메인 루틴 -----
7         th_id = thread_run(fn_th_func, loop=True)
8
9     # do something...
10    thread_stop(th_id) #쓰레드 종료

```

### 7.3.3 thread\_pause(th\_id)

#### 기능

Thread를 일시 정지합니다.

#### 인수

인수명	자료형	기본값	설명
th_id	int	-	일시 정지하고자 하는 thread ID

#### 리턴

값	설명
0	성공
음수값	실패

#### 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시

예외	설명
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

### 예제

```

1  def fn_th_func():
2      if check_motion()==0: # 수행 중인 모션이 없는 경우
3          set_digital_output(1, OFF)
4      else:
5          set_digital_output(1, ON)
6      #----- 메인 루틴 -----
7      th_id = thread_run(fn_th_func, loop=True)
8
9      # do something...
10
11     thread_pause(th_id) #쓰레드 일시 정지

```

## 7.3.4 thread\_resume(th\_id)

### 기능

일시 정지된 Thread를 다시 시작합니다.

### 인수

인수명	자료형	기본값	설명
th_id	int	-	일시 정지를 풀고 재개하고자 하는 thread ID

### 리턴

값	설명
0	성공
음수값	실패

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1  def fn_th_func():
2      if check_motion()==0: # 수행 중인 모션이 없는 경우
3          set_digital_output(1, OFF)
4      else:
5          set_digital_output(1, ON)
6
7      #----- 메인 루틴 -----
8      th_id = thread_run(fn_th_func, loop=True)
9
10     # do something...
11     thread_pause(th_id) #쓰레드 일시 정지
12
13     # do something...
14     thread_resume(th_id) #쓰레드 일시 정지 해제

```

### 7.3.5 thread\_state(th\_id)

#### 기능

Thread의 상태를 확인합니다.

#### 인수

인수명	자료형	기본값	설명
th_id	int	-	상태 정보를 알고자 하는 thread ID

## 리턴

값	설명
1	RUN (TH_STATE_RUN)
2	PAUSE (TH_STATE_PAUSE)
3	STOP (TH_STATE_STOP)

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시

## 예제

```

1 def fn_th_func():
2     if check_motion()==0: # 수행 중인 모션이 없는 경우
3         set_digital_output(1, OFF)
4     else:
5         set_digital_output(1, ON)
6
7     th_id = thread_run(fn_th_func, loop=True)
8     state1 = thread_state(th_id)
9
10    thread_pause(th_id)
11    state2 = thread_state(th_id)

```

### 7.3.6 통합 예제 - Thread

예제를 통한 쓰레드 사용법을 설명합니다

#### 예제 1: 쓰레드 예제

```

1 ----- thread 1: client comm. -----
2 def fn_th_client():
3     global g_sock
4     global g_cmd
5     res, rx_data = client_socket_read(g_sock)

```

```

6   if res > 0:
7     g_cmd = rx_data.decode() #decode: byte형을 string으로 변환
8   else: # 통신에러가 발생한 경우
9     client_socket_close(g_sock)
10    exit() #프로그램 종료
11    wait(0.1)
12    return 0
13
14  ----- thread 2: check IO -----
15  def fn_th_check_io():
16    if get_digital_input(1) == ON:
17      exit() #프로그램 종료
18      wait(0.1)
19      return 0
20
21  ----- main -----
22  g_sock = client_socket_open("192.168.137.2", 20002) #서버에 접속
23  g_cmd = ""
24
25  g_th_id1 = thread_run(fn_th_client, loop=True) # th_client 쓰레드 실행
26  g_th_id2 = thread_run(fn_th_check_io, loop=True) # th_check_io 쓰레드 실행
27
28  p1 = posj(0, 0, 90, 0, 90, 0)
29  p2 = posj(10, 0, 90, 0, 90, 0)
30  p3 = posj(20, 0, 90, 0, 90, 0)
31
32  while 1:
33    if g_cmd == "a":
34      g_cmd = ""
35      movej(p1, vel=100, acc=100)
36      client_socket_write(g_sock, b"end")
37    if g_cmd == "b":
38      g_cmd = ""
39      movej(p2, vel=100, acc=100)
40      client_socket_write(g_sock, b"end")
41    if g_cmd == "c":
42      g_cmd = ""
43      movej(p3, vel=100, acc=100)
44      client_socket_write(g_sock, b"end")
45      wait(0.1)

```

th\_client 쓰레드: server로부터 수신한 데이터를 string형으로 변하여 g\_cmd에 저장

th\_check\_io 쓰레드: 1번 접점의 상태를 체크하여 ON 이면 프로그램 종료

main: 서버에 접속,

1. 2개의 쓰레드 실행 : th\_client, th\_check\_io
2. 서버로부터 “a”가 수신되면 p1으로 이동 후, 서버에게 “end” 전송
3. 서버로부터 “b”가 수신되면 p2으로 이동 후, 서버에게 “end” 전송
4. 서버로부터 “c”가 수신되면 p3으로 이동 후, 서버에게 “end” 전송

## 7.4 기타

### 7.4.1 wait(time)

#### 기능

지정된 시간만큼 대기합니다.

#### 인수

인수명	자료형	기본값	설명
Time	Float	-	시간 [sec]

#### 리턴

값	설명
0	성공
음수값	오류

#### 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

#### 예제

```

1  wait(1.3) # 1.3초 대기
2
3  while 1: # 0.1초마다 접점 1번 상태를 체크
4    if get_digital_input(1) == ON:

```

5	set_digital_output(1, ON)
6	wait(0.1)

## 7.4.2 exit()

### 기능

현재 수행 중인 프로그램을 종료합니다.

### 리턴

값	설명
0	성공

### 예외

예외	설명
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시

### 예제

1	exit()
---	--------

## 7.4.3 sub\_program\_run(name)

### 기능

별도의 파일로 작성된 서브프로그램을 실행합니다.

### 인수

인수명	자료형	기본값	설명
name	string	-	서브 프로그램 이름

## 리턴

값	설명
module	실행된 서브 프로그램의 모듈 객체

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

### 알아두기

- 서브프로그램의 첫줄에 반드시 “from DRCF import \*” 구문이 있어야만 합니다.
- 티칭 펜던트 사용하는 경우, 본 구문이 자동 삽입됩니다.
- 메인프로그램과 서브프로그램간에 함수와 전역 변수를 서로 접근할 수 있습니다. 즉, 메인프로그램에서 서브프로그램의 함수와 전역 변수를 접근할 수 있고, 반대로 서브프로그램에서 메인프로그램의 함수와 전역 변수를 접근할 수 있습니다. 하기 예제2를 참조 바랍니다.
- 메인프로그램과 서브프로그램의 전역 변수 이름이 같은 경우, 각각 다른 변수로 동작함에 유의 바랍니다. 즉, 서로 변수 참조가 되지 않습니다.
- 메인프로그램과 서브프로그램의 함수 이름이 같은 경우, 스코프 를 상 호출된 파일내의 함수가 호출됩니다.

## 예제 1

```

1 # subProgramA, subProgramB 가 사전에 작성 저장되어 있어야 합니다.
2
3 <subProgramA.drl>
4 from DRCF import *
5 movej([0,0,90,0,90,0], vel=30, acc=30)
6
7 <subProgramB.drl>
8 from DRCF import *

```

```

9 movej[(10,0,90,0,90,0), vel=30, acc=30]
10
11 <main program>
12 while True:
13     var_select = tp_get_user_input("Select File", DR_VAR_INT)
14     if var_select == 0:
15         sub_program_run("subProgramA") # subProgramA 실행
16     elif var_select == 1:
17         sub_program_run("subProgramB") # subProgramB 실행

```

## 예제 2

```

1 # subProgram이 사전에 작성 저장되어 있어야 합니다.
2
3 <subProgram.drl>
4 from DRCF import *
5 g_sub_count = 1 # subProgram의 전역변수
6
7 def funcA():
8     global g_main_count
9     global g_main_var
10    g_main_var += g_main_count # main Program의 전역변수 접근
11    return 0
12
13 def funcB():
14     global g_sub_count
15     global g_main_var
16     g_main_var += g_sub_count # main Program의 전역변수 접근
17     return 0
18
19 global main_func
20 main_func() # main Program의 함수 접근
21
22 <main program>
23 def main_func(): # main Program의 함수
24     tp_log("main_func called")
25     return 0
26     g_main_var = 0 # main Program의 전역변수
27     g_main_count = 10 # main Program의 전역변수
28
29 #import subProgram
30 subModule = sub_program_run("subProgram")
31
32 #subProgram 내의 funcA() 함수 호출
33 subModule.funcA()
34
35 tp_log("funcA: g_main_var = " + str(g_main_var))
36
37 # subProgram 내의 전역 변수 g_sub_count 접근
38 subModule.g_sub_count += 1

```

```

39
40 #subProgram 내의 funcB() 함수 호출
41 subModule.funcB()
42
43 #연산 결과 값 : "12"
44 tp_log("funcA: g_main_var = " + str(g_main_var))

```

## 7.4.4 drl\_report\_line(option)

### 기능

DRL 스크립트 구동 시, 실행라인 표시 기능을 ON/OFF 할 수 있는 명령어입니다. 실행라인 표시 기능을 OFF 하면, 실행라인 표시 기능을 수행하기 위한 시간이 줄어들어 DRL 실행 속도가 현저히 빨라지게 됩니다.

#### ⚠ 주의

실행라인 표시 기능을 OFF 한 구간에서는 하기의 기능이 동작하지 않습니다.

- 라인 별 실행 시간 표시
- 변수 모니터링
- 시스템 변수 업데이트
- Debug 모드에서 Step by Step
- Debug 모드에서 Brake Point

### 인수

인수명	자료형	기본값	설명
option	Int	-	DRL 실행 라인 표시 여부 ON(1), OFF(0)

### 리턴

값	설명
없음	-

### 예제

```

1 x=0
2 y=0
3 drl_report_line(OFF) #실행라인 표시 기능 OFF
4 while x < 1000: #실행라인 표시 안 됨 (실행 속도 향상)
5   x += 1 #실행라인 표시 안 됨 (실행 속도 향상)

```

```

6   drl_report_line(ON) #실행라인 표시 기능 ON
7   x=0 #실행라인 표시됨
8   y=0 #실행라인 표시됨

```

## 7.4.5 set\_fm(key, value)

### 기능

기 정의되어 KT Smart Factory와 연동되고 있는 시스템 정보 이외에 프로그램 실행 시 생성된 변수(전역 변수 및 시스템 변수 등)정보에 대하여 연동이 필요한 경우에 사용하는 명령어입니다.



### 주의

Setup 메뉴의 KT Smart Factory 메뉴에서 연동 정보 미설정 시 기능이 동작하지 않음에 유의 바랍니다.  
KT Smart Factory 메뉴는 KT전용 라이선스 설정 시에만 표시됩니다

### 인수

인수명	자료형	기본값	설명
key	string	-	연동 데이터 이름
value	int float string	-	연동 데이터 변수 가능한 데이터 타입 <ul style="list-style-type: none"> <li>• 정수형 데이터</li> <li>• 실수형 데이터</li> <li>• 문자열 데이터</li> </ul>

### 리턴

값	설명
없음	-

### 예제

```

1 count = 0
2
3 movej(posj(0, 0, 90, 0, 90, 0), vel=30, acc=30)
4 while True:
5     movej(posj(0, 0, -90, 0, 90, 0), vel=30, acc=30)
6     movej(posj(0, 0, 90, 0, 90, 0), vel=30, acc=30)

```

```

7   count = count + 1
8   set_fm("TotalCount", count)

```

## 7.4.6 get\_robot\_model()

### 기능

로봇의 모델명을 읽는 명령어입니다.

### 리턴

값	설명
모델 명	String 타입으로 모델 정보를 리턴합니다. "M1013", "M0617", "M0609", "M1509", "A0307", "A0307S", "A0509", "A0509S", "A0912", "A0912S" "H2515", "H2017"

### 예제

```

1   model = get_robot_model()
2
3   if model == "M1013":
4       set_velj(30)
5   else:
6       set_velj(50)

```

## 7.4.7 get\_robot\_serial\_num()

### 기능

로봇의 일련번호를 읽는 명령어입니다.

### 리턴

값	설명
일련 번호	String 타입으로 일련번호를 리턴합니다. 일련번호: 숫자, 영문으로 구성된 6자리 문자열

## 예제

	1      serial_num = get_robot_serial_num()
--	--

## 7.4.8 check\_robot\_jts()

### 기능

로봇의 joint torque sensor 장착 여부를 확인하는 명령어입니다.

### 리턴

값	설명
0	성공
음수값	실패

## 예제

	1 <b>if</b> check_robot_jts() != True: 2        movej([0, 0, 90, 0, 90, 0], 60, 30) 3 <b>else:</b> 4        movej([0, 0, 0, 0, 0, 0], 60, 30)
--	--

## 7.4.9 check\_robot\_fts()

### 기능

로봇의 force torque sensor 장착 여부를 확인하는 명령어입니다.

### 리턴

값	설명
0	성공
음수값	실패

## 예제

```

1  if check_robot_fts() != True:
2      movej([0, 0, 90, 0, 90, 0], 60, 30)
3  else:
4      movej([0, 0, 0, 0, 0, 0], 60, 30)

```

## 7.4.10 start\_timer()

### 기능

제어기의 시뮬레이션 프로그램 수행시간을 측정하기 위한 명령어입니다. end\_timer() 명령어와 연동하여 두 함수의 사이에 있는 스크립트의 수행시간을 반환합니다.

#### **⚠ 주의**

해당 기능은 Windows 환경과 Linux 환경에서의 모션 수행 시간을 측정하기 위한 기능입니다. 에뮬레이터(가상 제어기) 환경에서 Real 모드로 측정 시 잘못된 값이 반환될 수 있습니다.

### 리턴

값	설명
0	성공
음수값	실패

## 예제

```

1  start_timer()
2  wait(1)
3  t= end_timer()
4  tp_log("tttt={0} sec".format(t))

```

### 관련 명령어

- [end\\_timer\(\)](#)(p. 302)

### 7.4.11 end\_timer()

#### 기능

제어기의 시뮬레이션 프로그램 수행시간을 측정하기 위한 명령어입니다. start\_timer() 명령어와 연동하여 두 함수의 사이에 있는 스크립트의 수행시간을 반환합니다.

**⚠ 주의**

해당 기능은 Windows 환경과 Linux 환경에서의 모션 수행 시간을 측정하기 위한 기능입니다. 에뮬레이터(가상 제어기) 환경에서 Real 모드로 측정 시 잘못된 값이 반환될 수 있습니다.

#### 리턴

값	설명
float	측정된 시간 정보(process 수행 시간)

#### 예제

```

1 start_timer()
2 wait(1)
3 t= end_timer()
4 tp_log("tttt={0} sec".format(t))

```

#### 관련 명령어

- [start\\_timer\(\)](#)(p. 301)

### 7.4.12 message\_to\_dp(event\_name, strdata)

#### 기능

사용자에게 메시지를 제공한 후, 사용자에게 응답 메세지를 전달 받으면 응답 메세지를 리턴합니다.

#### 인수

인수명	자료형	기본값	설명
event_name	string	-	사용자에게 제공할 event name • event_name은 256byte 이내로 제한됩니다.

인수명	자료형	기본값	설명
strdata	string	-	사용자에게 제공할 data <ul style="list-style-type: none"> <li>▪ strdata는 3000byte 이내로 제한됩니다.</li> </ul>

## 리턴

값	설명
string	sendUserEventResponse(Dart-API) 호출시 사용자가 입력한 event name
string	sendUserEventResponse(Dart-API) 호출시 사용자가 입력한 strdata

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 tp_log('message send!')
2 event, data = message_to_dp('wait', 'true')
3 tp_log('message get!')

```

## 7.4.13 send\_load\_module(package, sub\_type, active)

### 기능

Framework Module 을 load 하고 load 한 Module의 UniqueID를 리턴한다.

## 인수

인수명	자료형	기본값	설명
package	str	-	load 할 module 의 package name
sub_type	int	-	load 할 module 의 sub type 0 : None 1 : TCP/IP 2 : Serial 4 : Modbus Slave 5 : Force Compliance 7 : Modbus Master 10 : Other
active	int	-	load 후 active 여부 0 : no active 1 : active (sub_type 0  Force Compliance 인 경우만 사용)

## 리턴

값	설명
int	load 한 module 의 sub type
int	상태 값 0 : 성공 0 이외의 값 : 에러코드
int	load 한 module의 UniqueID
str	load 한 module의 package name

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시

예외	설명
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```
1 sub_type, status, id, name = send_load_module("com.dart.module.default.adm
ittancecontrol", 5, 1)
```

## 7.4.14 send\_unload\_module(unique\_id)

### 기능

Framework Module 을 unload 한다.

### 인수

인수명	자료형	기본값	설명
unique_id	int	-	unload 할 module의 UniqueID

### 리턴

값	설명
int	상태 값 0 : 성공 0 이외의 값 : 에러코드
int	unload 한 Module의 UniqueID

### 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시

예외	설명
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```
1 sub_type, status, id, name = send_load_module("com.dart.module.default.adm  
ittancecontrol", 5, 1)  
2 send_unload_module(id)
```

## 8 수학 함수

### 8.1 기본 함수

#### 8.1.1 ceil(x)

##### 기능

$x$ 값보다 크거나 같은 정수 중 가장 작은 정수 값을 리턴합니다. 단, 소수점 이하는 모두 올립니다.

##### 인수

인수명	자료형	기본값	설명
x	float	-	-

##### 리턴

값	설명
rounded integer	-

##### 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시

#### 8.1.2 floor(x)

##### 기능

$x$ 값보다 작거나 같은 정수 중 가장 큰 정수 값을 리턴합니다. 단, 소수점 이하는 모두 버립니다.

##### 인수

인수명	자료형	기본값	설명
x	float	-	-

## 리턴

값	설명
rounded integer	-

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시

## 8.1.3 pow(x, y)

### 기능

x의 y승을 한 값을 반환한다.

### 인수

인수명	자료형	기본값	설명
x	float	-	
y	float	-	

## 리턴

값	설명
x raised to the power y	-

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시

### 8.1.4 sqrt(x)

#### 기능

$x$ 의 제곱근을 리턴합니다.

#### 인수

인수명	자료형	기본값	설명
x	float	-	-

#### 리턴

값	설명
the square root of x	성공

#### 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시

### 8.1.5 log(x, b)

#### 기능

$x$ 의 로그를 밑수  $b$ 로 리턴합니다.

#### 인수

인수명	자료형	기본값	설명
x	float	-	-
b	float	-	base, e (natural logarithm)

## 리턴

값	설명
the logarithm of f to the base of b.	-

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시

## 8.1.6 d2r(x)

### 기능

x degree 값을 radians로 리턴합니다.

### 인수

인수명	자료형	기본값	설명
x	float	-	The angle in degrees

## 리턴

값	설명
The angle in radians	-

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시

## 8.1.7 r2d(x)

### 기능

x radians 값을 degree로 리턴합니다.

### 인수

인수명	자료형	기본값	설명
x	float		The angle in radians

### 리턴

값	설명
The angle in degrees	-

### 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시

## 8.1.8 random()

### 기능

0 이상 1 미만의 난수를 리턴합니다.

### 인수

인수명	자료형	기본값	설명

## 리턴

값	설명
random number	0 이상 1 미만의 난수(float)

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시

## 8.2 삼각함수

### 8.2.1 sin(x)

#### 기능

x radians의 sine 값을 리턴합니다.

#### 인수

인수명	자료형	기본값	설명
x	float	-	-

## 리턴

값	설명
the sine of x	-

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시

## 8.2.2 cos(x)

### 기능

$x$  radians의 cosine 값을 리턴합니다.

### 인수

인수명	자료형	기본값	설명
x	float	-	-

### 리턴

값	설명
the cosine of x	-

### 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시

## 8.2.3 tan(x)

### 기능

$x$  radians의 tangent 값을 리턴합니다.

### 인수

인수명	자료형	기본값	설명
x	float	-	-

## 리턴

값	설명
the tangent of x	-

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시

## 8.2.4 asin(x)

### 기능

x radians의 arc sine of 값 을 리턴합니다

### 인수

인수명	자료형	기본값	설명
x	float	-	

## 리턴

값	설명
the arc sine of x	-

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시

## 8.2.5 acos(x)

### 기능

$x$  radians의 arc cosine of 값을 리턴합니다.

### 인수

인수명	자료형	기본값	설명
x	float	-	-

### 리턴

값	설명
the arc cosine of $x$	-

### 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시

## 8.2.6 atan(x)

### 기능

$x$  radians의 arc tangent of 값을 리턴합니다.

### 인수

인수명	자료형	기본값	설명
x	float	-	-

## 리턴

값	설명
the arc tangent of x	-

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시

## 8.2.7 atan2(y, x)

### 기능

y/x radians의 arc tangent of 값을 리턴합니다.

### 인수

인수명	자료형	기본값	설명
y	float	-	-
x	float	-	-

## 리턴

값	설명
the arc tangent of y/x	The result is between -pi and pi

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시

## 8.3 선형대수

### 8.3.1 norm(x)

#### 기능

$x$ 의 L2 norm을 리턴합니다.

#### 인수

인수명	자료형	기본값	설명
$x$	float[3]	-	Point 좌표( $x, y, z$ )

#### 리턴

값	설명
float	Point 좌표 벡터의 크기

#### 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시

### 8.3.2 rotx(angle)

#### 기능

$x$ 축을 기준으로  $angle$  값만큼 회전시키는 회전행렬을 리턴합니다.

#### 인수

인수명	자료형	기본값	설명
$angle$	float	0	회전각 [deg]

## 리턴

값	설명
float[3][3]	회전행렬

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시

## 예제

```
1 rotm = rotx(30)
```

### 8.3.3 roty(angle)

#### 기능

y축을 기준으로 angle 값만큼 회전시키는 회전행렬을 리턴합니다.

#### 인수

인수명	자료형	기본값	설명
angle	float	0	회전각 [deg]

## 리턴

값	설명
float[3][3]	회전행렬

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시

**예제**

1	<code>rotm = roty(30)</code>
---	------------------------------

**8.3.4 rotz(angle)****기능**

`z`축을 기준으로 `angle` 값만큼 회전시키는 회전행렬을 리턴합니다.

**인수**

인수명	자료형	기본값	설명
<code>angle</code>	float	0	회전각 [deg]

**리턴**

값	설명
float[3][3]	회전행렬

**예외**

예외	설명
<code>DR_Error</code> ( <code>DR_ERROR_TYPE</code> )	인수들의 데이터형 오류 시

**예제**

1	<code>rotm = rotz(30)</code>
---	------------------------------

**8.3.5 rotm2eul(rotm)****기능**

회전행렬을 받아 Euler angle(`xyz` order)을 degree 값으로 리턴합니다. 단, 결과값으로 리턴되는 Euler angle(`rx`, `ry`, `rz`) 중 `ry` 값은 항상 양수로 리턴합니다.

**인수**

인수명	자료형	기본값	설명
rotm	Float[3][3]	-	회전행렬

**리턴**

값	설명
float[3]	ZYZ Euler angle [deg]

**예외**

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시

**예제**

```

1  rotm = [[1,0,0],[0,0.87,-0.5],[0,0.5,0.87]]
2
3  eul = rotm2eul(rotm)

```

**8.3.6 rotm2rotvec(rotm)****기능**

회전행렬을 받아 rotation vector(angle/axis representation)를 리턴합니다.

**인수**

인수명	자료형	기본값	설명
rotm	Float[3][3]	-	회전행렬

**리턴**

값	설명
float[3]	rotation vector [rad]

**예외**

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시

**예제**

```

1 rotm = [[1,0,0],[0,0.87,-0.5],[0,0.5,0.87]]
2
3 rotvec = rotm2rotvec(rotm)

```

**8.3.7 eul2rotm([alpha,beta,gamma])****기능**

Euler angle(zyz order)을 회전행렬로 변환합니다.

**인수**

인수명	자료형	기본값	설명
eul	float[3]	[0 0 0]	Euler angle (zyz) [deg]

**리턴**

값	설명
float[3][3]	회전행렬

**예외**

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시

**예제**

```

1 eul = [90, 90, 0]
2
3 rotm = eul2rotm (eul)

```

**8.3.8 eul2rotvec([alpha,beta,gamma])****기능**

Euler angle(zyz order)을 rotation vector로 변환합니다.

**인수**

인수명	자료형	기본값	설명
eul	float[3]	[0 0 0]	Euler angle (zyz) [deg]

**리턴**

값	설명
float[3]	rotation vector [rad]

**예외**

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시

## 예제

```

1 eul = [90, 90, 0]
2
3 rotvec = eul2rotvec (eul)

```

## 8.3.9 eul2rpy([alpha,beta,gamma])

### 기능

Euler angle zyz를 degree 값으로 입력 받아 euler angle z(=Yaw)y(=Pitch)x(=Roll)의 degree 값으로 리턴 합니다.

### 인수

인수명	자료형	기본값	설명
eul	float[3]	[0, 0, 0] [deg]	Euler angle (zyz) [deg] ※ alpha (A), beta (B), gamma (C) 순으로 입력

### 리턴

값	설명
float[3]	Euler angle (zyx) [deg] ※ yaw (Rz), pitch (Ry), roll (Rx) 순으로 return

### 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시

## 예제

```

1 eul = [10, 20, 30]
2 rpy = eul2rpy(eul)

```

### 8.3.10 rpy2eul([yaw,pitch,roll])

#### 기능

Euler angle z(=Yaw)y(=Pitch)x(=Roll)를 degree 값으로 입력 받아 euler angle zyz의 degree 값으로 리턴 합니다.

#### 인수

인수명	자료형	기본값	설명
rpy	float[3]	[0 0 0]	Euler angle (zyx) [deg] ※ yaw (Rz), pitch (Ry), roll (Rx) 순으로 입력

#### 리턴

값	설명
float[3]	Euler angle (zyz) [deg] ※ alpha(A), beta(B), gamma(C) 순으로 return

#### 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시

#### 예제

```

1   rpy = [10, 20, 30]
2   eul = rpy2eul(rpy)

```

### 8.3.11 rotvec2eul([rx,ry,rz])

#### 기능

Rotation vector를 Euler Angle(zyz)로 변환합니다.

**인수**

인수명	자료형	기본값	설명
rotvec	float[3]	-	rotation vector[rad]

**리턴**

값	설명
float[3]	ZYZ Euler angle [deg]

**예외**

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시

**예제**

```

1  rotvec = [0, 0, 0.7854]
2
3  eul = rotvec2eul(rotvec) # eul=[45,0,0]

```

**8.3.12 rotvec2rotm([rx,ry,rz])****기능**

rotation vector를 rotation matrix로 변환합니다.

**인수**

인수명	자료형	기본값	설명
rotvec	float[3]	-	rotation vector [rad]

## 리턴

값	설명
float[3][3]	회전행렬

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시

## 예제

```
1 rotm = rotvec2rotm([0.7854, 0, 0])
```

### 8.3.13 htrans(posx1, posx2, ori\_type\_out)

## 기능

posx1과 posx2로부터 구한 Homogeneous Transformation matrix를 T1, T2라 할 때 T1\*T2에 해당하는 자세를 리턴 합니다.

$$H_1 H_2 = \begin{bmatrix} R_1 & r_1 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} R_2 & r_2 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R_1 R_2 & r_1 + R_1 r_2 \\ 0 & 1 \end{bmatrix}$$

## 인수

인수명	자료형	기본값	설명
posx1	posx list (float[6])	-	posx 또는 position list [mm, deg]

인수명	자료형	기본값	설명
posx2	posx list (float[6])	-	posx 또는 position list [mm, deg]
ori_type_out	int	None	output orientation type <ul style="list-style-type: none"> <li>• None: pos 인자의 orientation type을 따름.</li> <li>• DR_ELR_ZYZ: Euler Angles(z-y'-z'', in degrees)</li> <li>• DR_ELR_ZYX: Euler Angles(z-y'-x'', in degrees)</li> <li>• DR_ELR_XYZ: Euler Angles(x-y'-z'', in degrees)</li> <li>• DR_FIX_XYZ: Fixed Angles(x-y-z, in degrees)</li> <li>• DR_ROTVEC: 3D rotation vector (angle/axis representation)</li> <li>• DR_QUAT: unit quaternion(x, y, z, w)</li> </ul>

## 리턴

값	설명
posx	[mm, deg]

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시

## 예제

```

1 posx1 = [100, 20, 300, 90, 0, 180]
2 posx2 = [200, 50, 100, 90, 30, 150]
3 posx = htrans(posx1, posx2)
4 posx_quat = htrans(posx1, posx2, DR_QUAT)
5
6
7

```

### 8.3.14 get\_intermediate\_pose(posx1, posx2, alpha, ori\_type\_out)

#### 기능

posx1으로부터 posx2로의 linear transition 중 alpha에 위치한 posx를 리턴합니다. alpha가 0이면 posx1을, alpha가 0.5이면 두 자세의 중간값을, alpha가 1이면 posx2를 리턴합니다.

#### 인수

인수명	자료형	기본값	설명
posx1	posx list (float[6])	-	posx 또는 position list [mm, deg]
posx2	posx list (float[6])	-	posx 또는 position list [mm, deg]
alpha	float	-	$0.0 \leq \text{alpha} \leq 1.0$
ori_type_out	int	None	output orientation type <ul style="list-style-type: none"> <li>• None: posx1 인자의 orientation type을 따름.</li> <li>• DR_ELR_ZYZ: Euler Angles(z-y'-z'', in degrees)</li> <li>• DR_ELR_ZYX: Euler Angles(z-y'-x'', in degrees)</li> <li>• DR_ELR_XYZ: Euler Angles(x-y'-z'', in degrees)</li> <li>• DR_FIX_XYZ: Fixed Angles(x-y-z, in degrees)</li> <li>• DR_ROTVEC: 3D rotation vector (angle/axis representation)</li> <li>• DR_QUAT: unit quaternion(x, y, z, w)</li> </ul>

#### 리턴

값	설명
posx	[mm, deg]

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시

## 예제

```

1 posx1 = [100, 20, 300, 90, 0, 180]
2 posx2 = [200, 50, 100, 90, 30, 150]
3 alpha = 0.5
4
5 posx = get_intermediate_pose(posx1,posx2,alpha)
6
7

```

### 8.3.15 get\_distance(posx1, posx2)

#### 기능

두 pose 포지션 사이의 거리를 [mm] 단위로 리턴합니다.

#### 인수

인수명	자료형	기본값	설명
posx1	posx list (float[6])	-	posx 또는 position list [mm]
posx2	posx list (float[6])	-	posx 또는 position list [mm]

#### 리턴

값	설명
float	[mm]

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시

## 예제

```

1 posx1 = [100, 20, 300, 90, 0, 180]
2
3 posx2 = [200, 50, 100, 90, 30, 150]
4
5 dis_posx = get_distance(posx1, posx2)

```

### 8.3.16 get\_normal(x1, x2, x3)

#### 기능

3개의 작업 공간 지점(posx)으로 이루어진 평면의 normal vector를 리턴합니다. 방향은 시계방향 기준입니다.

#### 인수

인수명	자료형	기본값	설명
x1	posx list (float[6])	-	posx 또는 position list
x2	posx list (float[6])	-	posx 또는 position list
x3	posx list (float[6])	-	posx 또는 position list

## 리턴

값	설명
float[3]	normal vector

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1  x1 = posx(0, 500, 700, 30, 0, 90)
2
3  x2 = posx(500, 0, 700, 0, 0, 45)
4
5  x3 = posx(300, 100, 500, 45, 0, 45)
6
7  vect = get_normal(x1, x2, x3)

```

### 8.3.17 add\_pose(posx1,posx2,ori\_type\_out)

## 기능

두 자세의 합을 구합니다.

$$\text{add\_pose}\left(\begin{bmatrix} R_1 & r_1 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} R_2 & r_2 \\ 0 & 1 \end{bmatrix}\right) \Rightarrow \begin{bmatrix} R_1R_2 & r_1 + r_2 \\ 0 & 1 \end{bmatrix}$$

## 인수

인수명	자료형	기본값	설명
posx1	posx list (float[6])	-	posx 또는 position list [mm, deg]
posx2	posx list (float[6])	-	posx 또는 position list [mm, deg]
ori_type_out	int	None	output orientation type <ul style="list-style-type: none"> <li>• None: posx1 인자의 orientation type을 따름.</li> <li>• DR_ELR_ZYZ: Euler Angles(z-y'-z'', in degrees)</li> <li>• DR_ELR_ZYX: Euler Angles(z-y'-x'', in degrees)</li> <li>• DR_ELR_XYZ: Euler Angles(x-y'-z'', in degrees)</li> <li>• DR_FIX_XYZ: Fixed Angles(x-y-z, in degrees)</li> <li>• DR_ROTVEC: 3D rotation vector (angle/axis representation)</li> <li>• DR_QUAT: unit quaternion(x, y, z, w)</li> </ul>

## 리턴

값	설명
posx	[mm, deg]

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시

## 예제

```
1 posx1 = [100, 20, 300, 90, 0, 180]
```

```

2 posx2 = [200, 50, 100, 90, 30, 150]
3
4 add_posx = add_pose(posx1, posx2)
5
6 add_posx_zyx = add_pose(posx1, posx2, DR_ELR_ZYX)
7

```

### 8.3.18 subtract\_pose(posx1, posx2, ori\_type\_out)

#### 기능

두 자세의 차이를 구합니다.

$$\text{subtract\_pose}\left(\begin{bmatrix} R_1 & r_1 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} R_2 & r_2 \\ 0 & 1 \end{bmatrix}\right) \Rightarrow \begin{bmatrix} R_2^T R_1 & r_1 - r_2 \\ 0 & 1 \end{bmatrix}$$

#### 인수

인수명	자료형	기본값	설명
posx1	posx list (float[6])	-	posx 또는 position list [mm, deg]
posx2	posx list (float[6])	-	posx 또는 position list [mm, deg]
ori_type_out	int	None	output orientation type <ul style="list-style-type: none"> <li>• None: posx1 인자의 orientation type을 따릅니다.</li> <li>• DR_ELR_ZYZ: Euler Angles(z-y'-z'', in degrees)</li> <li>• DR_ELR_ZYX: Euler Angles(z-y'-x'', in degrees)</li> <li>• DR_ELR_XYZ: Euler Angles(x-y'-z'', in degrees)</li> <li>• DR_FIX_XYZ: Fixed Angles(x-y-z, in degrees)</li> <li>• DR_ROTVEC: 3D rotation vector (angle/axis representation)</li> <li>• DR_QUAT: unit quaternion(x, y, z, w)</li> </ul>

## 리턴

값	설명
posx	[mm, deg]

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시

## 예제

```

1 posx1 = [100, 20, 300, 90, 0, 180]
2 posx2 = [200, 50, 100, 90, 30, 150]
3 subtract_posx = subtract_pose(posx1, posx2)
4 subtract_posx_zyx = subtract_pose(posx1, posx2, DR_ELR_ZYX)

```

## 8.3.19 inverse\_pose(posx1,ori\_type\_out)

### 기능

posx의 inverse에 해당하는 posx 값을 리턴합니다.

$$\text{inv\_pose} \cdot \begin{bmatrix} R_1 & r_1 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R_1 & r_1 \\ 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} R_1^T & -R_1^T r_1 \\ 0 & 1 \end{bmatrix}$$

## 인수

인수명	자료형	기본값	설명
posx1	posx list (float[6])	-	posx 또는 position list [mm, deg]

인수명	자료형	기본값	설명
ori_type_out	int	None	<p>output orientation type</p> <ul style="list-style-type: none"> <li>• None: pos1 인자의 orientation type을 따름.</li> <li>• DR_ELR_ZYZ: Euler Angles(z-y'-z'', in degrees)</li> <li>• DR_ELR_ZYX: Euler Angles(z-y'-x'', in degrees)</li> <li>• DR_ELR_XYZ: Euler Angles(x-y'-z'', in degrees)</li> <li>• DR_FIX_XYZ: Fixed Angles(x-y-z, in degrees)</li> <li>• DR_ROTVEC: 3D rotation vector (angle/axis representation)</li> <li>• DR_QUAT: unit quaternion(x, y, z, w)</li> </ul>

## 리턴

값	설명
posx	[mm, deg]

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시

## 예제

```

1 posx1 = [100, 20, 300, 90, 0, 180]
2
3 inv_posx = inverse_pose(posx1)

```

## 8.3.20 dot\_pose(posx1, posx2)

### 기능

두 개의 pose가 주어졌을 때 translation 성분의 내적을 구합니다.

## 인수

인수명	자료형	기본값	설명
posx1	posx list (float[6])	-	posx 또는 position list [mm, deg]
posx2	posx list (float[6])	-	posx 또는 position list [mm, deg]

## 리턴

값	설명
float	두 개 pose의 내적 결과

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시

## 예제

```

1 posx1 = [100, 20, 300, 90, 0, 180]
2 posx2 = [200, 50, 100, 90, 30, 150]
3 res= dot_pose(posx1, posx2)

```

## 8.3.21 cross\_pose(posx1, posx2)

### 기능

두 개의 pose가 주어졌을 때 translation 성분의 외적을 구합니다.

## 인수

인수명	자료형	기본값	설명
posx1	posx list (float[6])	-	posx 또는 position list [mm, deg]
posx2	posx list (float[6])	-	posx 또는 position list [mm, deg]

## 리턴

값	설명
float[3]	두 개 pose의 외적 결과

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시

## 예제

```

1 posx1 = [100, 20, 300, 90, 0, 180]
2
3 posx2 = [200, 50, 100, 90, 30, 150]
4
5 res= cross_pose(posx1, posx2)

```

## 8.3.22 unit\_pose(posx1)

### 기능

주어진 posx translation 성분의 단위 백터를 구합니다.

## 인수

인수명	자료형	기본값	설명
posx1	posx list (float[6])	-	posx 또는 position list [mm, deg]

## 리턴

값	설명
float[3]	주어진 posx의 단위 백터

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시

## 예제

```

1 posx1 = [100, 20, 300, 90, 0, 180]
2
3 res = unit_pose(posx1)

```

## 9 외부 통신 명령어

### 9.1 Serial

#### 9.1.1 serial\_open(**port=None, baudrate=115200, bytesize=DR\_EIGHTBITS, parity=DR\_PARITY\_NONE, stopbits=DR\_STOPBITS\_ONE**)

##### 기능

Serial 통신 포트를 오픈합니다.

##### 인수

인수명	자료형	기본값	설명
port	string	None	<ul style="list-style-type: none"> <li>D-SUB(9 pin) 연결 : "COM"</li> <li>USB to Serial 연결 : "COM_USB"</li> </ul>
baudrate	int	115200	Baud rate 2400, 4800, 9600, 19200, 38400, 57600, 115200
bytesize	int	8	데이터 bit 수 <ul style="list-style-type: none"> <li>DR_FIVEBITS: 5</li> <li>DR_SIXBITS: 6</li> <li>DR_SEVENBITS: 7</li> <li>DR_EIGHTBITS: 8</li> </ul>
parity	str	"N"	Parity checking <ul style="list-style-type: none"> <li>DR_PARITY_NONE: "N"</li> <li>DR_PARITY EVEN: "E"</li> <li>DR_PARITY ODD: "O"</li> <li>DR_PARITY MARK: "M"</li> <li>DR_PARITY SPACE: "S"</li> </ul>
stopbits	int	1	Stop bit의 수 <ul style="list-style-type: none"> <li>DR_STOPBITS_ONE = 1</li> <li>DR_STOPBITS_ONE_POINT_FIVE = 1.5</li> <li>DR_STOPBITS_TWO = 2</li> </ul>

## 리턴

값	설명
serial.Serial instance	연결 성공

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	Serial.SerialException 예외 발생

## 예제

```

1 # 시리얼 포트 D-SUB(9 pin)에 연결한 경우
2 ser = serial_open(port="COM", baudrate=115200, bytesize=DR_EIGHTBITS,
3 parity=DR_PARITY_NONE, stopbits=DR_STOPBITS_ONE)
4
5 res = serial_write(ser, b"123ABC")
6
7 serial_close(ser)
8
9
10 # USB to serial 장비를 USB 포트에 연결한 경우
11 ser = serial_open(port="COM_USB", baudrate=115200, bytesize=DR_EIGHTBITS,
12 parity=DR_PARITY_NONE, stopbits=DR_STOPBITS_ONE)
13
14 res = serial_write(ser, b"123ABC")
15
16 serial_close(ser)

```

### 9.1.2 serial\_close(ser)

#### 기능

Serial 통신 포트를 닫습니다.

## 인수

인수명	자료형	기본값	설명
ser	serial.Serial	-	Serial instance

## 리턴

값	설명
0	Serial 포트 닫기 성공

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수의 데이터형 오류 시

## 예제

```

1   ser = serial_open(port="COM", baudrate=115200, bytesize=DR_EIGHTBITS,
2     parity=DR_PARITY_NONE, stopbits=DR_STOPBITS_ONE)
3
4   res = serial_write(serial, b"123456789")
5
6   serial_close(serial)

```

## 9.1.3 serial\_state(*ser*)

### 기능

Serial 통신 포트의 상태를 리턴합니다.

## 인수

인수명	자료형	기본값	설명
ser	serial.Serial	-	Serial instance

## 리턴

값	설명
1	Serial 포트 opened 상태
0	Serial 포트 closed 상태

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수의 데이터형 오류 시

## 예제

```

1 ser = serial_open(port="COM", baudrate=115200, bytesize=DR_EIGHTBITS,
2                         parity=DR_PARITY_NONE, stopbits=DR_STOPBITS_ONE)
3
4 state = serial_state(ser)
5
6 serial_close(ser)

```

## 9.1.4 serial\_set\_inter\_byte\_timeout(**ser, timeout=None**)

### 기능

포트에 읽기/쓰기를 수행할 때 바이트 간(inter-byte)의 timeout을 설정합니다.

### 인수

인수명	자료형	기본값	설명
ser	serial.Serial	-	Serial instance
timeout	float	None	읽기/쓰기 수행 시, 바이트 간의 timeout <ul style="list-style-type: none"> <li>• timeout 발생 이전까지 처리된 데이터를 연속적인 데일터로 처리</li> <li>• None : inter-byte timeout을 지정하지 않음</li> </ul>

## 리턴

값	설명
0	성공

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수의 데이터형 오류 시

## 예제

```

1 ser = serial_open(port="COM", baudrate=115200, bytesize=DR_EIGHTBITS,
2     parity=DR_PARITY_NONE, stopbits=DR_STOPBITS_ONE)
3
4 res = serial_set_inter_byte_timeout(ser, 0.1)
5
6 serial_close(ser)

```

## 9.1.5 serial\_write(*ser*, *tx\_data*)

### 기능

Serial 포트에 데이터(*tx\_data*)를 기록합니다.

### 인수

인수명	자료형	기본값	설명
<i>ser</i>	serial.Serial	-	Serial instance
<i>tx_data</i>	byte	-	송신할 데이터 • 데이터 타입 byte 형이어야 합니다. • 하기 예제 참조 바랍니다.

## 리턴

값	설명
0	성공
-1	Port가 open 상태가 아닙니다.
-2	serial.SerialException 예외 발생

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수의 데이터형 오류 시

## 예제

```

1 ser = serial_open(port="COM", baudrate=115200, bytesize=DR_EIGHTBITS,
2     parity=DR_PARITY_NONE, stopbits=DR_STOPBITS_ONE)
3
4 serial_write(ser, b"123456789") #b는 bytes 형을 의미합니다.
5
6 # string을 byte 형으로 변환
7 msg = "abcd"                  # msg는 string 변수
8 serial_write(ser, msg.encode()) # encode()는 string형을 byte형으로 변환
9
10 serial_close(ser)

```

## 9.1.6 serial\_read(ser, length=-1, timeout=-1)

### 기능

Serial 포트에서 데이터를 읽어옵니다.

### 인수

인수명	자료형	기본값	설명
Ser	serial.Serial	-	Serial instance

인수명	자료형	기본값	설명
Length	int	-1	Read 할 바이트 수 • -1: 미지정(read 된 데이터만큼 읽습니다.) • n(>=0): 지정한 바이트 수만큼 읽습니다.
timeout	int float	-1	Read 대기 시간 • -1: 무한 대기 • n(>0): n 초(second)

## 리턴

값(res, rx_data)	설명	
res	n	수신한 데이터의 바이트 수
	-1	Port가 open 상태가 아닙니다.
	-2	serial.SerialException 예외 발생
rx_data		Read 한 데이터(byte type)

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시

## 예제

```

1  ser = serial_open(port="COM", baudrate=115200, bytesize=DR_EIGHTBITS,
2      parity=DR_PARITY_NONE, stopbits=DR_STOPBITS_ONE)
3
4  res, rx_data = serial_read(ser)
5  #데이터가 수신될 때 까지 무한 대기
6
7  res, rx_data = serial_read(ser, timeout=3)
8  #데이터가 수신될 때 까지 대기, 3초 타임 아웃 설정
9  #3초 이내 수신되면 읽은 데이터 바로 리턴

```

```

10 #3초가 지나면 현재까지 읽은 값을 리턴한다
11
12 res, rx_data = serial_read(ser, length=100)
13 #100byte 읽을 때까지 무한대기
14
15 res, rx_data = serial_read(ser, length=100, timeout=3)
16 #100byte 읽을 때 까지 대기, 3초 타임 아웃 설정
17 #3초 이내 100byte 수신되면 읽은 데이터 바로 리턴
18 #3초가 지나면 현재까지 읽은 값을 리턴한다
19
20 #수신된 byte형을 string 형으로 변환
21 rx_msg = rx_data.decode()
22 #rx_data는 byte형이고 string형으로 변환하기 위해서는 decode() 사용합니다.
23 #예를 들어, rx_data = b"abcd" 이면, rx_msg="abcd"가 됩니다.
24
25 res, rx_data = serial_close(ser)

```

## 9.1.7 serial\_get\_count()

### 기능

연결된 USB to Serial의 포트정보, 장치이름을 읽어옵니다.

### 리턴

값(port_info, device_name)	설명
count	연결된 시리얼 포트 개수

### 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시

### 예제

```

1 count = serial_get_count() # 연결된 시리얼 포트 개수 읽기
2
3 for i in range(count):
4     port_info, device_name = serial_get_info(i+1)

```

5

```
tp_popup("i={}, port ={}, dev ={}".format(i, port_info, device_name))
```

## 9.1.8 serial\_get\_info(id)

### 기능

연결된 USB to Serial의 포트정보, 장치이름을 읽어옵니다.

### 인수

인수명	자료형	기본값	설명
id	int	1	읽고싶은 USB to Serial의 ID (1~10)

### 리턴

값(port_info, device_name)	설명
port_info	포트정보 (값이 NULL인 경우 연결된 장치 없음)
device_name	장치이름 (값이 NULL인 경우 연결된 장치 없음)

### 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시

### 예제

```

1  port_info, device_name = serial_get_info(1) #1번째 연결된 장치의 정보조회
2  #port_info = "COM_USB"
3  ser = serial_open(port=port_info, baudrate=115200, bytesize=DR_EIGHTBITS,
4    parity=DR_PARITY_NONE, stopbits=DR_STOPBITS_ONE)

```

## 9.1.9 통합 예제 - Serial

serial 포트에서 RXD(2번pin) 와 TXD(3번pin) 결선한 후, 자체 loop-back 테스트하는 예제입니다.

## 예제 1 : 자체 loop-back test 예제

```

1 # 시리얼 포트 OPEN
2 # D-SUB(9pin) 연결한 경우 : port="COM"
3 # USB to Serial로 USB에 연결한 경우 : port="COM_USB"
4 ser = serial_open(port="COM_USB", baudrate=115200, bytesize=DR_EIGHTBITS,
5 parity=DR_PARITY_NONE, stopbits=DR_STOPBITS_ONE)
6 wait(1)

7 # SEND DATA : "123ABC"
8 res = serial_write(ser, b"123ABC") # b는 바이트 타입을 의미
9 wait(1)

10 # READ DATA
11 res, rx_data = serial_read(ser)
12 # H/W 적으로 RXD, TXD가 연결되어 있어 res=6(바이트) rx_data = b"123ABC" 가 수신 됨
13
14 tp_popup("res ={0}, rx_data={1}".format(res, rx_data))

15 # 해당 시리얼 포트를 닫음
16 serial_close(ser)
17
18

```

송신 데이터를 그대로 수신데이터로 받아서 결과를 TP pop-up 메시지로 출력합니다.

정상적으로 동작한 경우 res=6 rx\_data = b"123ABC" 결과를 출력합니다.

## 예제 2 : 다양한 패킷 전송 예제

송신 패킷 : "MEAS\_START" +data1[4byte]+data2[4byte]

data1: integer를 4byte로 변환 ex) 1 → 00000001

data2: integer를 4byte로 변환 ex) 2 → 00000002

ex) data1=1, data2=2 인 경우: "MEAS\_START"+00000001+00000002

실제 패킷: 4D4541535F53544152540000000100000002

수신 패킷 : res=18, rx\_data="MEAS\_START"+00000001+00000002

rxd1 추출 : 10~14번째 byte를 interger로 변환

rxd2 추출 : 14~18번째 byte를 interger로 변환

```

1 ser = serial_open(port="COM_USB", baudrate=115200, bytesize=DR_EIGHTBITS,
2 parity=DR_PARITY_NONE, stopbits=DR_STOPBITS_ONE)
3 wait(1)

4 send_data = b"MEAS_START" # b는 바이트 타입을 의미
5 data1 = 1

```

```

6   data2 =2
7   send_data += (data1).to_bytes(4, byteorder='big')
8   send_data += (data2).to_bytes(4, byteorder='big')
9
10  # SEND DATA
11  res = serial_write(serv, send_data)
12  wait(1)
13
14  # READ DATA
15  # H/W 적으로 RXD, TXD가 연결되어 있어 send_data 가 그대로 수신 됨
16  res, rx_data = serial_read(serv)
17
18  tp_popup("res ={0}, rx_data={1}".format(res, rx_data))
19
20  rxd1 = int.from_bytes(rx_data[10:10+4], byteorder='big', signed=True)
21  rxd2 = int.from_bytes(rx_data[14:14+4], byteorder='big', signed=True)
22
23  tp_popup("res={0}, rxd1={1}, rxd2={2}".format(res, rxd1, rxd2))
24
25
26  #해당 시리얼 포트를 닫음
27  serial_close(serv)

```

USB to serial 장비를 USB 포트에 연결하고 byte형 send\_data를 전송합니다.

송신 데이터를 그대로 수신하도록 RXD(2pin) 와 TXD(3pin)를 결선하였으므로

res = 18, rx\_data는 send\_data와 같은 패킷을 가집니다.

rxd1 추출 : 10~14번째 byte를 interger로 변환

rxd2 추출 : 14~18번째 byte를 interger로 변환

최종 결과는 res=18, rxd1=1, rxd2=2 가 됩니다.

### 9.1.10 플랜지 I/O

`flange_serial_open(baudrate=115200, bytesize=DR_EIGHTBITS, parity=DR_PARITY_NONE, stopbits=DR_STOPBITS_ONE)`

#### 기능

Pseudo Flange Serial 통신 포트를 열기 위한 명령어입니다.

Pseudo Flange Serial 통신은 일반 Serial 통신과 일부 다른 특성이 있습니다. 따라서 Handshaking 방식의 통신으로 사용하는 것을 권장합니다. (예, Modbus RTU) 센서등에 사용한다면 내부 버퍼 크기의 제한(255bytes)과 내부 딜레이로 인해 overflow가 발생할 수 있습니다.

(단, 신형 Flange I/O ( M/H Series 2024.03.22 & A/E Series 2024.04.11 이후 ) 에서는 지원하지 않으며, Robot Parameter 모듈의 Flange I/O 탭에서 설정 가능합니다.)

**인수**

인수명	자료형	기본값	설명
baudrate	int	115200	Baud rate 2400, 4800, 9600, 19200, 38400, 57600, 115200 etc
bytesize	int	8	데이터 bit 수 <ul style="list-style-type: none"> <li>• DR_FIVEBITS : 5</li> <li>• DR_SIXBITS : 6</li> <li>• DR_SEVENBITS : 7</li> <li>• DR_EIGHTBITS : 8</li> </ul>
parity	str	"N"	Parity checking <ul style="list-style-type: none"> <li>• DR_PARITY_NONE: "N"</li> <li>• DR_PARITY EVEN: "E"</li> <li>• DR_PARITY ODD: "O"</li> </ul>
stopbits	int	1	Stop bit의 수 <ul style="list-style-type: none"> <li>• DR_STOPBITS_ONE =1</li> <li>• DR_STOPBITS_TWO =2</li> </ul>

**리턴**

값	설명
0	성공
음수값	실패

**예외**

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIN E)	C Extension 모듈 에러 발생 시

예외	설명
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

**flange\_serial\_close()****기능**

Pseudo Flange Serial을 닫기 위한 명령어입니다.

(단, 신형 Flange I/O ( M/H Series 2024.03.22 & A/E Series 2024.04.11 이후 )에서는 지원하지 않으며, Robot Parameter 모듈의 Flange I/O탭에서 설정 가능합니다.)

**리턴**

값	설명
0	성공

**예외**

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

**flange\_serial\_write(tx\_data, port = 1)****기능**

Pseudo Flange Serial에 data를 쓰기 위한 명령어입니다.

**인수**

인수명	자료형	기본값	설명
tx_data	byte	-	송신할 데이터(최대32byte) • 데이터 타입 byte 형이어야 합니다.
port	int	1	write 하고자 하는 포트번호 X1 포트 : 1 X2 포트 : 2 (A모델 사용 불가)

**리턴**

값	설명
0	성공

**예외**

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

**flange\_serial\_read(timeout=None, port = 1)**

**기능**

Pseudo Flange Serial에서 data를 읽어오기 위한 명령어입니다.

### 인수

인수명	자료형	기본값	설명
timeout	float int	None	Read 대기시간
port	int	1	read 하고자 하는 포트번호 X1 포트 : 1 X2 포트 : 2 (A모델 사용 불가)

### 리턴

값	설명
res	수신한 데이터의 바이트 수 -1 : time out -2: overflow
rx_data	Read 한 데이터

### 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

### 통합 예제 - Serial(플랜지 I/O)

예제 1) Pseudo Flange Serial을 이용하여 Robotiq 2F를 제어하는 예제

```

1  Def recv_data():
2      Res, data = flange_serial_read(3) #timeout 3s
3      If res == -1:

```

```

4 #Exception Handling Required
5     tp_log("Response time out!!")
6 elif res == -2
7 #Exception Handling Required
8     tp_log("Buffer Overflow!!")
9 else :
10 if Modbus_recv_check(data) == True:
11     tp_log("recv size [“+str(res)+”] :” +str(data))
12 else
13 #Exception Handling Required
14     tp_log("CRC Check Fail!!")
15
16 flange_serial_open(baudrate=115200, bytesize=DR_EIGHTBITS, parity=DR_PARITY_NONE, stopbits = DR_STOPBITS_ONE)
17 wait(0.1)
18
19 #Step1:Activation Request(clear Act)
20 flange_serial_write(modbus_send_make(b"\x09\x10\x03\xE8\x00\x03\x06\x00\x00\x00\x00\x00\x00"))
21 res, data = flange_serial_read()
22
23 #Step1:Activation Request(set Act)
24 flange_serial_write(modbus_send_make(b"\x09\x10\x03\xE8\x00\x03\x06\x01\x00\x00\x00\x00\x00"))
25 res, data = flange_serial_read()
26
27 #Step 2: Read Gripper status until the activation is completed
28 flange_serial_write(modbus_send_make(b"\x09\x03\x07\xD0\x00\x01"))
29 res, data = flange_serial_read()
30
31 #Step 3: Move the robot to the pick-up location
32 wait(1)
33
34 #Step 4: Close the Gripper at full speed and full force
35 flange_serial_write(modbus_send_make(b"\x09\x10\x03\xE8\x00\x03\x06\x09\x00\xFF\xFF\xFF"))
36 res, data = flange_serial_read()
37
38 #Step 5: Read Gripper status until the grasp is completed
39 flange_serial_write(modbus_send_make(b"\x09\x03\x07\xD0\x00\x03"))
40 res, data = flange_serial_read()
41
42 #Step 6: Move the robot to the release location
43 wait(1)
44
45 #Step 7: Open the Gripper at full speed and full force
46 flange_serial_write(modbus_send_make(b"\x09\x10\x03\xE8\x00\x03\x06\x09\x00\xFF\xFF\xFF"))
47 res, data = flange_serial_read()
48
49 #Step 8: Read Gripper status until the opening is completed
50 flange_serial_write(modbus_send_make(b"\x09\x03\x07\xD0\x00\x03"))
51 res, data = flange_serial_read()
52

```

53	flange_serial_close()
----	-----------------------

### 예제 2) 신규 Flange Serial을 이용하여 onRobot RC6 V2를 제어하는 예제

```

1 # select port (1,2)
2 # If your robot is A, you can select 1 only
3 port = 1
4
5 tp_log("==>START OPEN==>")
6 flange_serial_write(modbus_send_make(b"\x41\x06\x00\x00\x01\x2c"), port)
7 wait(1)
8 res, data = flange_serial_read(1,port )
9 wait(1)
10
11 flange_serial_write(modbus_send_make(b"\x41\x06\x00\x01\x03\xe8"), port )
12 wait(1)
13 res, data = flange_serial_read(1,1)
14 wait(1)
15
16 flange_serial_write(modbus_send_make(b"\x41\x06\x00\x02\x00\x01"), port )
17 wait(1)
18 res, data = flange_serial_read(1,port )
19 wait(1)
20
21
22 tp_log("==>START CLOSE==>")
23 flange_serial_write(modbus_send_make(b"\x41\x06\x00\x00\x01\x2c"), port )
24 wait(1)
25 res, data = flange_serial_read(1,port )
26 wait(1)
27
28 flange_serial_write(modbus_send_make(b"\x41\x06\x00\x01\x00\x00"), port )
29 wait(1)
30 res, data = flange_serial_read(1,port )
31 wait(1)
32
33 flange_serial_write(modbus_send_make(b"\x41\x06\x00\x02\x00\x01"), port )
34 wait(1)
35 res, data = flange_serial_read(1,port )
36 wait(1)

```

## 9.2 Tcp/Client

### 9.2.1 client\_socket\_open(ip, port)

#### 기능

로봇 제어기가 클라이언트 소켓을 생성하고, 서버(ip, port)에 연결을 시도합니다.

서버에 연결되면, 연결된 소켓을 반환합니다.

#### 인수

인수명	자료형	기본값	설명
ip	str	-	Server IP address: (예) “192.168.137.200”
port	int	-	Server Port number (예) 20002

#### 리턴

값	설명
socket.socket instance	연결 성공

#### 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	연결 처리 중 socket.error 예외 발생

## 예제

```

1  sock = client_socket_open("192.168.137.200", 20002)
2  # 서버(ip="192.168.137.200", port=20002)에 무한 접속 시도를 합니다.
3  # 정상적으로 연결된 경우, 연결된 소켓을 반환합니다.
4  # 반환된 소켓(sock)을 이용하여 하기와 같이 데이터를 쓰고 읽고 닫습니다.
5
6  client_socket_write(sock, b"123abc")    #서버에 데이터전송 (b는 byte형을 의미)
7  res, rx_data = client_socket_read(sock) #서버로부터 데이터 수신
8  client_socket_close(sock)             #서버와의 연결 닫기

```

## 9.2.2 client\_socket\_close(sock)

### 기능

서버와의 통신을 종료합니다. 서버에 다시 접속하려면 반드시 client\_socket\_close(sock)로 소켓을 닫고, 재 open 해야 합니다.

### 인수

인수명	자료형	기본값	설명
sock	socket.socket	-	client_socket_open()에서 리턴 받은 socket instance

### 리턴

값	설명
0	연결 종료 성공

### 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수의 데이터형 오류 시
DR_Error (DR_ERROR_RUNTIME)	종료 처리 중 socket.error 예외 발생

## 예제

```

1  sock = client_socket_open("192.168.137.200", 20002)
2  # 서버(ip="192.168.137.200", port=20002)에 무한 접속 시도를 합니다.
3
4  # do something ...
5
6  client_socket_close(sock) #서버와의 통신을 종료합니다.

```

### 9.2.3 client\_socket\_state(sock)

#### 기능

소켓의 정상 상태 여부를 리턴합니다. Server와의 접속 상태를 알기 위해서는 client\_socket\_read 혹은 client\_socket\_write의 리턴값으로 확인합니다. (예제2 참조)

#### 인수

인수명	자료형	기본값	설명
sock	socket.socket	-	client_socket_open()에서 리턴 받은 socket instance

#### 리턴

값	설명
1	소켓 정상 상태
0	소켓 비정상 상태

#### 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수의 데이터형 오류 시

#### 예제

```

1  sock = client_socket_open("192.168.137.200", 20002)
2

```

```

3 state = client_socket_state(sock) #소켓의 상태를 읽습니다.
4
5 client_socket_close(sock)

```

## 예제 2

```

1 sock = client_socket_open("192.168.137.200", 20002)
2
3 res, rx_data =client_socket_read(sock)
4 tp_log("[RX] res={0}, rx_data ={1}".format(res, rx_data))
5 if (res < 0):
6     tp_log("[RX] server disconnect")      # 서버와의 통신이 끊어 진 경우
7     client_socket_close(sock)
8     exit()
9
10 client_socket_close(sock)

```

### 9.2.4 client\_socket\_write(sock, tx\_data)

#### 기능

서버에게 데이터를 송신합니다.

#### 인수

인수명	자료형	기본값	설명
sock	socket.socket	-	client_socket_open()에서 리턴 받은 socket instance
tx_data	byte	-	송신할 데이터 <ul style="list-style-type: none"> <li>• 데이터 타입이 byte 형이어야 합니다.</li> <li>• 하기 예제 참조 바랍니다.</li> </ul>

#### 리턴

값	설명
0	성공
-1	Server와 연결된 상태가 아닙니다.
-2	Server와 연결이 끊어졌거나, 송신 처리 중 socket.error 예외 발생

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수의 데이터형 오류 시

## 예제

```

1  sock = client_socket_open("192.168.137.200", 20002)
2
3  client_socket_write(sock, b"1234abcd") #b 는 byte 형을 의미합니다.
4
5  # string 을 byte 형으로 변환
6  msg = "abcd"                      # msg는 string 변수
7  client_socket_write(sock, msg.encode()) # encode()는 string형을 byte형으로 변환
8
9  client_socket_close(sock)

```

## 9.2.5 client\_socket\_read(sock, length=-1, timeout=-1)

### 기능

서버로부터 데이터를 수신합니다.

### 인수

인수명	자료형	기본값	설명
sock	socket.socket	-	client_socket_open()에서 리턴 받은 socket instance
length	int	-1	수신할 데이터의 바이트 수 <ul style="list-style-type: none"> <li>-1 : 미지정 (수신된 데이터만큼 읽습니다.)</li> <li>n(&gt;=0) : 지정한 바이트 수만큼 읽습니다.</li> </ul>
timeout	int float	-1	수신 대기 시간 <ul style="list-style-type: none"> <li>-1 : 무한 대기</li> <li>n(&gt;0) : n 초(second)</li> </ul>

## 리턴

값(res, rx_data)		설명
res	>0	수신한 데이터의 바이트 수
	-1	Server와 연결된 상태가 아닙니다.
	-2	수신 처리 중 socket.error 예외 발생
	-3	지정한 수신 timeout 발생
rx_data		수신한 데이터(byte 형)

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시

## 예제

```

1  sock = client_socket_open("192.168.137.200", 20002)
2
3  res, rx_data = client_socket_read(sock)      # 데이터를 수신할때까지 무한 대기
4  # length 생략했으므로 수신된 데이터만큼 읽고,
5  # timeout 생략했으므로 수신될 때 까지 무한 대기 합니다.
6  # 데이터가 수신되면 (res = 수신 데이터 사이즈, rx_data=수신데이터) 반환됩니다.
7
8  res, rx_data = client_socket_read(sock, timeout=3) # 데이터 수신 될까지 3초간 대기
9  # 데이터가 3초안에 수신되면 (res = 수신 데이터 사이즈, rx_data=수신데이터) 반환
10 # 데이터가 3초안에 수신 안되면 (res = -3, rx_data=None) 반환
11
12 res, rx_data = client_socket_read(sock, length=64)    #수신 데이터 64바이트 읽기
13
14 res, rx_data = client_socket_read(sock, length=64, timeout=3)
15 # 3초 타임 아웃을 가지고 수신 데이터 64바이트 읽기
16
17 #수신된 byte형을 string 형으로 변환
18 rx_msg = rx_data.decode() #rx_data는 byte형이고 string형으로 변환하기 위해서는

```

```

19         #decode() 사용합니다.
20         #예를 들어, rx_data = b"abcd" 이면,
21         #rx_msg="abcd"가 됩니다.
22
23     client_socket_close(sock)

```

## 9.2.6 통합 예제 (Tcp/Client)

server IP = 192.168.137.200, open port =20002 이고,  
수신한 패킷을 그대로 client에게 전송하는 동작(mirroring)을 한다고 가정합니다

### 예제 1: 기본 Tcp Client 예제

```

1  # server IP = 192.168.137.200, open port =20002 로 가정
2  g_sock = client_socket_open("192.168.137.200", 20002)
3
4  tp_popup("connect O.K!",DR_PM_MESSAGE)
5  while 1:
6      client_socket_write(g_sock, b"abcd")  # byte형으로 스트링 "abcd" 전송
7      wait(0.1)
8      res, rx_data = client_socket_read(g_sock) #server로부터 수신 대기
9      tp_popup("res={0}, rx_data ={1}".format(res, rx_data), DR_PM_MESSAGE)
10     wait(0.1)

```

server에 접속하고 string "abcd"를 전송합니다. (b는 string을 byte형으로 변환합니다.)

server로부터 수신된 메시지를 TP에 출력합니다.

server는 수신된 데이터를 그대로 송신하므로 res = 4, rx\_data=b"abcd" 가 됩니다.

### 예제 2: 다양한 패킷 전송 예제

송신 패킷 : "MEAS\_START" +data1[4byte]+data2[4byte]

- data1: integer를 4byte로 변환 ex) 1 → 00000001
- data2: integer를 4byte로 변환 ex) 2 → 00000002

ex) data1=1, data2=2 인 경우: "MEAS\_START"+00000001+00000002

- 실제 패킷: 4D4541535F53544152540000000100000002
- 수신 패킷 : res=18, rx\_data="MEAS\_START"+00000001+00000002
  - rxd1 추출 : 10~14번째 byte를 interger로 변환
  - rxd2 추출 : 14~18번째 byte를 interger로 변환

```

1  g_sock = client_socket_open("192.168.137.100", 20002)
2  tp_popup("connect O.K!",DR_PM_MESSAGE)

```

```

3     send_data = b"MEAS_START"
4     data1 = 1
5     data2 = 2
6     send_data += (data1).to_bytes(4, byteorder='big')
7     send_data += (data2).to_bytes(4, byteorder='big')
8
9
10    client_socket_write(g_sock, send_data)
11
12    wait(0.1)
13
14    res, rx_data = client_socket_read(g_sock)
15    tp_popup("res={0}, rx_data ={1}".format(res, rx_data), DR_PM_MESSAGE)
16
17    rxd1 = int.from_bytes(rx_data[10:10+4], byteorder='big', signed=True)
18    rxd2 = int.from_bytes(rx_data[14:14+4], byteorder='big', signed=True)
19
20    tp_popup("res={0}, rxd1={1}, rxd2={2}".format(res, rxd1, rxd2),
21              DR_PM_MESSAGE)
22
23    client_socket_close(g_sock)

```

server에 접속하고 byte형 send\_data를 전송합니다.

server는 수신된 데이터를 그대로 송신하므로 res = 18, rx\_data는 send\_data와 같은 패킷을 가집니다.

- rxd1 추출 : 10~14번째 byte를 interger로 변환
- rxd2 추출 : 14~18번째 byte를 interger로 변환

최종 결과는 res=18, rxd1=1, rxd2=2 가 됩니다.

### 예제 3 : 재 접속

```

1  def fn_reconnect():
2      global g_sock
3      client_socket_close(g_sock)
4      g_sock = client_socket_open("192.168.137.200", 20002)
5      return
6
7      g_sock = client_socket_open("192.168.137.200", 20002)
8      tp_popup("connect O.K!", DR_PM_MESSAGE)
9
10     client_socket_write(g_sock, b"abcd")
11     wait(0.1)
12
13     while 1:
14         res, rx_data = client_socket_read(g_sock)
15         if res < 0:
16             fn_reconnect()
17         else:
18             tp_popup("res={0}, rx_data ={1}".format(res, rx_data), DR_PM_MESSAGE)

```

19

wait(0.1)

client\_socket\_read() 명령어의 리턴값을 체크합니다.

server와 접속이 끊어지거나, 통신 문제가 발생하면 음수 값이 리턴됩니다.

음수값이 리턴된 경우 reconnect() 함수를 호출하여 재접속을 시도합니다.

재 접속 시도 시, 기존에 open된 소켓을 닫는 것에 유의 바랍니다.

## 9.3 Tcp/Server

### 9.3.1 server\_socket\_open(port)

#### 기능

로봇 제어기가 서버 소켓을 생성하고, client의 연결을 대기합니다. client가 연결되면, 연결된 소켓을 반환합니다.

#### 인수

인수명	자료형	기본값	설명
port	int	-	오픈 할 port 번호

#### 리턴

값	설명
socket.socket instance	연결 성공

#### 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	연결 처리 중 socket.error 예외 발생

## 예제

```

1  sock = server_socket_open(20002)
2  # 20002 port를 열고 client가 접속해 올 때까지 대기합니다.
3  # 정상적으로 연결된 경우, 연결된 소켓을 반환합니다.
4  # 반환된 소켓(sock)을 이용하여 하기와 같이 데이터를 쓰고 읽고 닫습니다.
5
6  server_socket_write(sock, b"123abc") #client에 데이터 전송 (b는 byte형을 의미)
7  res, rx_data = server_socket_read(sock) #client로부터 데이터 수신
8
9  server_socket_close(sock)           # client와의 연결 닫기

```

### 9.3.2 server\_socket\_close(sock)

#### 기능

client 와의 통신을 종료합니다. client와 재 접속하려면 반드시 server\_socket\_close(sock)로 연결 종료 후, 재 open 해야합니다.

#### 인수

인수명	자료형	기본값	설명
sock	socket.socket	-	server_socket_open()에서 리턴 받은 socket instance

#### 리턴

값	설명
0	연결 종료 성공

#### 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수의 데이터형 오류 시
DR_Error (DR_ERROR_RUNTIM E)	종료 처리 중 socket.error 예외 발생

## 예제

```

1  sock = server_socket_open(20002)
2  # 20002 port를 열고 client가 접속해 올 때까지 대기합니다.
3
4  # do something ...
5
6  server_socket_close(sock) ) #client 와의 통신을 종료합니다.

```

### 9.3.3 server\_socket\_state(sock)

#### 기능

소켓의 정상 상태 여부를 리턴합니다.

Client 와의 접속 상태를 알기 위해서는 server\_socket\_read 혹은 server\_socket\_write의 리턴값으로 확인합니다. ( 예제2 참조)

#### 인수

인수명	자료형	기본값	설명
sock	socket.socket	-	server_socket_open()에서 리턴 받은 socket instance

#### 리턴

값	설명
1	소켓 정상 상태
0	소켓 비정상 상태

#### 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수의 데이터형 오류 시

#### 예제 1

```
1  sock = server_socket_open(20002)
```

```

2 state = server_socket_state(sock) #소켓의 상태를 읽습니다.
3
4
5 server_socket_close(sock)

```

## 예제 2

```

1 sock = server_socket_open(20002)
2
3 res, rx_data =server_socket_read(sock)
4 tp_log("[RX] res={0}, rx_data ={1}".format(res, rx_data))
5 if (res < 0):    #클라이언트 접속이 끊어진 경우
6     tp_log("[RX] client disconnect")
7     server_socket_close(sock)
8     exit()
9
10 server_socket_close(sock)

```

### 9.3.4 server\_socket\_write(sock, tx\_data)

#### 기능

Client에 데이터를 송신합니다.

#### 인수

인수명	자료형	기본값	설명
sock	socket.socket	-	server_socket_open()에서 리턴 받은 socket instance
tx_data	byte	-	송신할 데이터 <ul style="list-style-type: none"> <li>데이터 타입 byte 형이어야 합니다.</li> <li>하기 예제 참조 바랍니다.</li> </ul>

#### 리턴

값	설명
0	성공
-1	Client 와 연결된 상태가 아닙니다.
-2	Client 와 연결이 끊어졌거나, 송신 처리 중 socket.error 예외 발생

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수의 데이터형 오류 시

## 예제

```

1  sock = server_socket_open(20002)
2
3  server_socket_write(sock, b"1234abcd") #b 는 byte 형을 의미합니다.
4
5  # string 을 byte 형으로 변환
6  msg = "abcd"                      # msg는 string 변수
7  server_socket_write(sock, msg.encode()) # encode()는 string형을 byte형으로 변
8  환
9
server_socket_close(sock)

```

### 9.3.5 server\_socket\_read(sock, length=-1, timeout=-1)

#### 기능

Client로부터 데이터를 수신합니다.

#### 인수

인수명	자료형	기본값	설명
sock	socket.socket	-	server_socket_open()에서 리턴 받은 socket instance
length	int	-1	<p>수신할 데이터의 바이트 수</p> <ul style="list-style-type: none"> <li>-1 : 미지정 (수신된 데이터만큼 읽습니다.)</li> <li>n(&gt;=0) : 지정한 바이트 수만큼 읽습니다.</li> </ul>
timeout	int float	-1	<p>수신 대기 시간</p> <ul style="list-style-type: none"> <li>-1 : 무한 대기</li> <li>n(&gt;0) : n 초(second)</li> </ul>

## 리턴

값(res, rx_data)	설명
res	0 수신한 데이터의 바이트 수
	-1 Client와 연결된 상태가 아닙니다.
	-2 수신 처리 중 socket.error 예외 발생
	-3 지정한 수신 timeout 발생
rx_data	수신한 데이터(byte 형)

## 예제

```

1  sock = server_socket_open(20002)
2
3  res, rx_data = server_socket_read(sock)      # 데이터를 수신할때까지 무한 대기
4  # length 생략했으므로 수신된 데이터만큼 읽고,
5  # timeout 생략했으므로 수신될 때 까지 무한 대기 합니다.
6  # 데이터가 수신되면 (res = 수신 데이터 사이즈, rx_data=수신데이터) 반환됩니다.
7
8  res, rx_data = server_socket_read(sock, timeout=3) #데이터 수신될까지 3초간 대기
9  # 데이터가 3초안에 수신되면 (res = 수신 데이터 사이즈, rx_data=수신데이터) 반환
10 # 데이터가 3초안에 수신 안되면 (res = -3, rx_data=None) 반환
11
12 res, rx_data = server_socket_read(sock, length=64) #수신 데이터 64바이트 읽기
13
14 res, rx_data = server_socket_read(sock, length=64, timeout=3)
15 # 3초 타임 아웃을 가지고 수신 데이터 64바이트 읽기
16
17 #수신된 byte형을 string 형으로 변환
18 rx_msg = rx_data.decode() #rx_data는 byte형이고 string형으로 변환하기 위해서는
19 #decode() 사용합니다.
20 #예를 들어, rx_data = b"abcd" 이면,
21 #rx_msg="abcd"가 됩니다.
22
23 server_socket_close(sock)

```

### 9.3.6 통합 예제 - Tcp/Server

client는 제어기 IP = 192,168,137.100, port = 20002로 접속을 하고

수신한 패킷을 그대로 server에게 전송하는 동작(mirroring)을 한다고 가정합니다.

## 예제 1: 기본 Tcp Server 예제

```

1 g_sock = server_socket_open(20002)
2 tp_popup("connect O.K!", DR_PM_MESSAGE)
3
4 while 1:
5     server_socket_write(g_sock, b"abcd") # byte형으로 스트링 "abcd" 전송
6     wait(0.1)
7     res, rx_data = server_socket_read(g_sock) #server로부터 수신 대기
8     tp_popup("res={0}, rx_data ={1}".format(res, rx_data), DR_PM_MESSAGE)
9     wait(0.1)

```

port=20002를 열고 client가 접속할 때까지 기다립니다.

접속된 client에 접속하고 byte형 “abcd”를 전송합니다.

client로 부터 수신된 메시지를 TP에 출력합니다.

client는 수신된 데이터를 그대로 송신하므로 res = 4, rx\_data=b"abcd" 가 됩니다.

## 예제 2: 다양한 패킷 전송 예제

송신 패킷 : “MEAS\_START” +data1[4byte]+data2[4byte]

data1: integer를 4byte로 변환 ex) 1 → 00000001

data2: integer를 4byte로 변환 ex) 2 → 00000002

ex) data1=1, data2=2 인 경우: “MEAS\_START”+00000001+00000002

실제 패킷: 4D4541535F3544152540000000100000002

수신 패킷 : res=18, rx\_data=“MEAS\_START”+00000001+00000002

rx1 추출 : 10~14번째 byte를 interger로 변환

rx2 추출 : 14~18번째 byte를 interger로 변환

```

1 g_sock = server_socket_open(20002)
2 tp_popup("connect O.K!", DR_PM_MESSAGE)
3
4 send_data = b"MEAS_START"
5 data1 =1
6 data2 =2
7 send_data += (data1).to_bytes(4, byteorder='big')
8 send_data += (data2).to_bytes(4, byteorder='big')
9
10 server_socket_write(g_sock, send_data)
11
12 wait(0.1)
13
14 res, rx_data = server_socket_read(g_sock)

```

```

15 tp_popup("res={0}, rx_data ={1}").format(res, rx_data), DR_PM_MESSAGE)
16
17 rxd1 = int.from_bytes(rx_data[10:10+4], byteorder='big', signed=True)
18 rxd2 = int.from_bytes(rx_data[14:14+4], byteorder='big', signed=True)
19
20 tp_popup("res={0}, rxd1={1}, rxd2={2}").format(res, rxd1, rxd2),
21 DR_PM_MESSAGE)
22 server_socket_close(g_sock)

```

client에 byte형 send\_data를 전송합니다.

client는 수신된 데이터를 그대로 송신하므로 res = 18, rx\_data는 send\_data와 같은 패킷을 가집니다.

rxd1 추출 : 10~14번째 byte를 interger로 변환

rxd2 추출 : 14~18번째 byte를 interger로 변환

최종 결과는 res=18, rxd1=1, rxd2=2 가 됩니다.

### 예제 3: 재 접속

```

1 def fn_reopen():
2     global g_sock
3     server_socket_close(g_sock)
4     g_sock = server_socket_open(20002)
5     return
6
7     g_sock = server_socket_open(20002)
8     tp_popup("connect O.K!",DR_PM_MESSAGE)
9
10    server_socket_write(g_sock, b"abcd")
11    wait(0.1)
12
13    while 1:
14        res, rx_data = server_socket_read(g_sock)
15        if res < 0:
16            fn_reopen()
17        else:
18            tp_popup("res={0}, rx_data ={1}").format(res, rx_data), DR_PM_MESSAGE)
19            wait(0.1)

```

server\_socket\_read() 명령어의 리턴값을 체크합니다.

client 와 접속이 끊어지거나, 통신 문제가 발생하면 음수 값이 리턴됩니다.

음수값이 리턴된 경우 reopen() 함수를 호출하여 client 접속을 기다립니다.

재 접속 open 시, 기존에 open된 소켓을 닫는 것에 유의 바랍니다.

## 9.4 Modbus

### 9.4.1 add\_modbus\_signal(ip, port, name, reg\_type, index, value=0, slaveid=255)

#### 기능

ModbusTCP의 신호를 등록합니다. Modbus I/O 설정의 경우 티치펜던트 I/O set-up 메뉴에서 설정해야하지만 티치펜던트 사용이 어려운 경우에 테스트를 위해서만 본 명령어를 사용하시기 바랍니다. 이 명령어를 사용하여 셋팅한 경우 티치펜던트에서 Modbus 관련 메뉴가 동작하지 않습니다.

#### 인수

인수명	자료형	기본값	설명
ip	string	-	modbusTCP 모듈 ip 주소
port	int	-	modbusTCP 모듈 port
name	string	-	modbus signal 이름
reg_type	int	-	Modbus의 신호 타입 <ul style="list-style-type: none"> <li>• DR_MODBUS_DIG_INPUT</li> <li>• DR_MODBUS_DIG_OUTPUT</li> <li>• DR_MODBUS_REG_INPUT</li> <li>• DR_MODBUS_REG_OUTPUT</li> </ul>
index	int	-	Modbus signal의 index
value	int	0	type0  DR_MODBUS_DIG_OUTPUT 또는 DR_MODBUS_REG_OUTPUT일 때 출력값 (그 외 경우에는 무시됩니다.)
slaveid	int	255	<ul style="list-style-type: none"> <li>• Slave ID of the ModbusTCP module (0 or 1-247 or 255)</li> <li>0 : Broadcast address</li> <li>255 : Default value for ModbusTCP</li> </ul>

## 리턴

값	설명
0	성공
음수값	실패

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 #Modbus IO 2개를 연결하고 접점을 할당하는 예제
2 #Modbus IO 1번 : IP 192.168.127.254, input 8점: "di1"~"di8", output 8점:
3   "do1"~"do8"
4 #Modbus IO 2번 : IP 192.168.127.253, input 8점: "di9"~"di16", output 8점:
5   "do9"~"do16"
6
7   # set <modbus 1> input : di1~di8
8   add_modbus_signal(ip="192.168.127.254",port=502, name="di1", reg_type=DR_M
9     ODBUS_REG_INPUT, index=0)
10  add_modbus_signal(ip="192.168.127.254",port=502, name="di2", reg_type=DR_M
11    ODBUS_REG_INPUT, index=1)
12  add_modbus_signal(ip="192.168.127.254",port=502, name="di3", reg_type=DR_M
13    ODBUS_REG_INPUT, index=2)
14  add_modbus_signal(ip="192.168.127.254",port=502, name="di4", reg_type=DR_M
15    ODBUS_REG_INPUT, index=3)
16  add_modbus_signal(ip="192.168.127.254",port=502, name="di5", reg_type=DR_M
17    ODBUS_REG_INPUT, index=4)

```

```

11 add_modbus_signal(ip="192.168.127.254",port=502, name="di6", reg_type=DR_M
12 ODBUS_REG_INPUT, index=5)
13 add_modbus_signal(ip="192.168.127.254",port=502, name="di7", reg_type=DR_M
14 ODBUS_REG_INPUT, index=6)
15 add_modbus_signal(ip="192.168.127.254",port=502, name="di8", reg_type=DR_M
16 ODBUS_REG_INPUT, index=7)
17
18 # set <modbus 1> output : do1~do8
19 add_modbus_signal(ip="192.168.127.254",port=502, name="do1", reg_type=DR_M
20 ODBUS_REG_OUTPUT, index=0, value=0)
21 add_modbus_signal(ip="192.168.127.254",port=502, name="do2", reg_type=DR_M
22 ODBUS_REG_OUTPUT, index=1, value=0)
23 add_modbus_signal(ip="192.168.127.254",port=502, name="do3", reg_type=DR_M
24 ODBUS_REG_OUTPUT, index=2, value=0)
25 add_modbus_signal(ip="192.168.127.254",port=502, name="do4", reg_type=DR_M
26 ODBUS_REG_OUTPUT, index=3, value=0)
27 add_modbus_signal(ip="192.168.127.254",port=502, name="do5", reg_type=DR_M
28 ODBUS_REG_OUTPUT, index=4, value=0)
29 add_modbus_signal(ip="192.168.127.254",port=502, name="do6", reg_type=DR_M
30 ODBUS_REG_OUTPUT, index=5, value=0)
31 add_modbus_signal(ip="192.168.127.254",port=502, name="do7", reg_type=DR_M
32 ODBUS_REG_OUTPUT, index=6, value=0)
33 add_modbus_signal(ip="192.168.127.254",port=502, name="do8", reg_type=DR_M
34 ODBUS_REG_OUTPUT, index=7, value=0)
35
36 ##### =====
37 # set <modbus 2> input : di9~di16
38 add_modbus_signal(ip="192.168.127.253",port=502, name="di9", reg_type=DR_
39 MODBUS_REG_INPUT, index=0)
40 add_modbus_signal(ip="192.168.127.253",port=502, name="di10", reg_type=DR_
41 MODBUS_REG_INPUT, index=1)
42 add_modbus_signal(ip="192.168.127.253",port=502, name="di11", reg_type=DR_
43 MODBUS_REG_INPUT, index=2)
44 add_modbus_signal(ip="192.168.127.253",port=502, name="di12", reg_type=DR_
45 MODBUS_REG_INPUT, index=3)
46 add_modbus_signal(ip="192.168.127.253",port=502, name="di13", reg_type=DR_
47 MODBUS_REG_INPUT, index=4)
48 add_modbus_signal(ip="192.168.127.253",port=502, name="di14", reg_type=DR_
49 MODBUS_REG_INPUT, index=5)
50 add_modbus_signal(ip="192.168.127.253",port=502, name="di15", reg_type=DR_
51 MODBUS_REG_INPUT, index=6)
52 add_modbus_signal(ip="192.168.127.253",port=502, name="di16", reg_type=DR_
53 MODBUS_REG_INPUT, index=7)
54
55 # set <modbus 2> output : do9~do16
56 add_modbus_signal(ip="192.168.127.253",port=502, name="do9", reg_type=DR_
57 MODBUS_REG_OUTPUT, index=0, value=0)
58 add_modbus_signal(ip="192.168.127.253",port=502, name="do10", reg_type=DR_
59 MODBUS_REG_OUTPUT, index=1, value=0)
60 add_modbus_signal(ip="192.168.127.253",port=502, name="do11", reg_type=DR_
61 MODBUS_REG_OUTPUT, index=2, value=0)
62 add_modbus_signal(ip="192.168.127.253",port=502, name="do12", reg_type=DR_
63 MODBUS_REG_OUTPUT, index=3, value=0)

```

```

41 add_modbus_signal(ip="192.168.127.253",port=502, name="do13", reg_type=DR_
42 MODBUS_REG_OUTPUT, index=4, value=0)
43 add_modbus_signal(ip="192.168.127.253",port=502, name="do14", reg_type=DR_
44 MODBUS_REG_OUTPUT, index=5, value=0)
45 add_modbus_signal(ip="192.168.127.253",port=502, name="do15", reg_type=DR_
46 MODBUS_REG_OUTPUT, index=6, value=0)
47 add_modbus_signal(ip="192.168.127.253",port=502, name="do16", reg_type=DR_
48 MODBUS_REG_OUTPUT, index=7, value=0)

```

### 9.4.2 add\_modbus\_rtu\_signal(slaveid=1, port=None, baudrate=115200, bytesize=DR\_EIGHTBITS, parity=DR\_PARITY\_NONE, stopbits=DR\_STOPBITS\_ONE, name, reg\_type, index, value=0)

#### 기능

ModbusRTU의 신호를 등록합니다. Modbus I/O 설정의 경우 티치펜던트 I/O set-up 메뉴에서 설정해야하지만 티치펜던트 사용이 어려운 경우에 테스트를 위해서만 본 명령어를 사용하시기 바랍니다. 이 명령어를 사용하여 셋팅한 경우 티치펜던트에서 Modbus 관련 메뉴가 동작하지 않습니다.

#### 인수

인수명	자료형	기본값	설명
slaveid	int	1	<ul style="list-style-type: none"> <li>ModbusRTU의 Slave ID입력(0 또는 1-247)0 : Broadcast address</li> </ul>
port	string	None	<ul style="list-style-type: none"> <li>D-SUB(9 pin) 연결 : "COM"</li> <li>USB to Serial 연결 : "COM_USB"</li> </ul>
baudrate	int	115200	Baud rate 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200
bytesize	int	8	데이터 bit 수 <ul style="list-style-type: none"> <li>DR_EIGHTBITS: 8</li> </ul>
parity	string	"N"	Parity checking <ul style="list-style-type: none"> <li>DR_PARITY_NONE: "N"</li> <li>DR_PARITY EVEN: "E"</li> <li>DR_PARITY ODD: "O"</li> </ul>
stopbits	int	1	Stop bit의 수 <ul style="list-style-type: none"> <li>DR_STOPBITS_ONE =1</li> <li>DR_STOPBITS_TWO =2</li> </ul>

인수명	자료형	기본값	설명
name	string	-	modbus signal 이름
reg_type	int	-	Modbus의 신호 타입 <ul style="list-style-type: none"> <li>• DR_MODBUS_DIG_INPUT</li> <li>• DR_MODBUS_DIG_OUTPUT</li> <li>• DR_MODBUS_REG_INPUT</li> <li>• DR_MODBUS_REG_OUTPUT</li> </ul>
index	int	-	Modbus signal의 index
value	int	0	type이 DR_MODBUS_DIG_OUTPUT 또는 DR_MODBUS_REG_OUTPUT일 때 출력값 (그 외 경우에는 무시됩니다.)

## 리턴

값	설명
0	성공
음수값	실패

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 add_modbus_rtu_signal(slaveid=1, port=port_info, baudrate=115200, bytesize=D
R_EIGHTBITS, parity=DR_PARITY_NONE, stopbits=DR_STOPBITS_ONE, name='d1',
reg_type=DR_MODBUS_REG_INPUT, index=0)
2 add_modbus_rtu_signal(slaveid=1, port=port_info, baudrate=115200, bytesize=D
R_EIGHTBITS, parity=DR_PARITY_NONE, stopbits=DR_STOPBITS_ONE, name='d01',
reg_type=DR_MODBUS_REG_OUTPUT, index=0, value=12345)

```

### 9.4.3 add\_modbus\_signal\_multi(ip, port, slaveid=255, name=None, reg\_type=DR\_HOLDING\_REGISTER, start\_address=0, cnt=1)

#### 기능

ModbusTCP의 FC15, FC16 사용을 위한 다중신호를 등록합니다. Modbus I/O 설정의 경우 티치펜던트 I/O set-up 메뉴에서 설정해야하지만 티치펜던트 사용이 어려운 경우에 테스트를 위해서만 본 명령어를 사용하시기 바랍니다. 이 명령어를 사용하여 셋팅한 경우 티치펜던트에서 Modbus 관련 메뉴가 동작하지 않습니다.

#### 알아두기

초기값 설정기능을 지원하지 않습니다.

#### 인수

인수명	자료형	기본값	설명
ip	string	-	modbusTCP 모듈 ip 주소
port	int	-	modbusTCP 모듈 port
slaveid	int	255	<ul style="list-style-type: none"> <li>Slave ID of the ModbusTCP module (0 or 1-247 or 255)</li> <li>0 : Broadcast address</li> <li>255 : Default value for ModbusTCP</li> </ul>
name	string	None	modbus signal 이름

인수명	자료형	기본값	설명
reg_type	int	DR_HOLDING_REGISTER	Modbus의 신호 타입 • DR_COIL = DR_MODBUS_DIG_OUTPUT • DR_HOLDING_REGISTER = DR_MODBUS_REG_OUTPUT
start_address	int	0	Modbus signal의 시작 주소
cnt	int	1	Modbus signal의 개수(최대 50개)

## 리턴

값	설명
0	성공
음수값	실패

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```
1 add_modbus_signal_multi(ip="192.168.137.200", port=502, slaveid=255, name="multi", reg_type=DR_HOLDING_REGISTER, start_address=0, cnt=5)
```

#### 9.4.4 add\_modbus\_rtu\_signal\_multi(slaveid=1, port=None, baudrate=115200, bytesize=DR\_EIGHTBITS, parity=DR\_PARITY\_NONE, stopbits=DR\_STOPBITS\_ONE, name=None, reg\_type=DR\_HOLDING\_REGISTER, start\_address=0, cnt=1)

##### 기능

ModbusRTU의 FC15, FC16 사용을 위한 다중신호를 등록합니다. Modbus I/O 설정의 경우 티치펜던트 I/O set-up 메뉴에서 설정해야하지만 티치펜던트 사용이 어려운 경우에 테스트를 위해서만 본 명령어를 사용하시기 바랍니다. 이 명령어를 사용하여 셋팅한 경우 티치펜던트에서 Modbus 관련 메뉴가 동작하지 않습니다.

##### 알아두기

- 초기값 설정기능을 지원하지 않습니다.

##### 인수

인수명	자료형	기본값	설명
slaveid	int	1	<ul style="list-style-type: none"> <li>ModbusRTU의 Slave ID입력(0 또는 1-247)</li> <li>0 : Broadcast address</li> </ul>
port	string	None	<ul style="list-style-type: none"> <li>D-SUB(9 pin) 연결 : "COM"</li> <li>USB to Serial 연결 : "COM_USB"</li> </ul>
baudrate	int	115200	<p>Baud rate 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200</p>
bytesize	int	8	<p>데이터 bit 수</p> <ul style="list-style-type: none"> <li>DR_EIGHTBITS: 8</li> </ul>
parity	string	"N"	<p>Parity checking</p> <ul style="list-style-type: none"> <li>DR_PARITY_NONE: "N"</li> <li>DR_PARITY_EVEN: "E"</li> <li>DR_PARITY_ODD: "O"</li> </ul>
stopbits	int	1	<p>Stop bit의 수</p> <ul style="list-style-type: none"> <li>DR_STOPBITS_ONE = 1</li> <li>DR_STOPBITS_TWO = 2</li> </ul>
name	string	-	modbus signal 이름

인수명	자료형	기본값	설명
reg_type	int	DR_HOLDING_REGISTER	Modbus의 신호 타입 • DR_COIL = DR_MODBUS_DIG_OUTPUT • DR_HOLDING_REGISTER = DR_MODBUS_REG_OUTPUT
start_address	int	-	Modbus signal의 시작주소
cnt	int	1	Modbus signal의 개수(최대 50개)

## 리턴

값	설명
0	성공
음수값	실패

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```
1 add_modbus_rtu_signal_multi(slaveid=1, port="COM", baudrate=115200,
bytesize=DR_EIGHTBITS, parity=DR_PARITY_NONE, stopbits=DR_STOPBITS_ONE,
name="multi", reg_type=DR_HOLDING_REGISTER, start_address=0, cnt=5)
```

## 9.4.5 del\_modbus\_signal(name)

### 기능

등록된 Modbus의 신호를 삭제합니다. Modbus I/O 설정의 경우 티치펜던트 I/O set-up 메뉴에서 설정해야 하지만 티치펜던트 사용이 어려운 경우에 테스트를 위해서만 본 명령어를 사용하시기 바랍니다 이 명령어를 사용하여 셋팅한 경우 티치펜던트에서 Modbus 관련 메뉴가 동작하지 않습니다.

### 인수

인수명	자료형	기본값	설명
name	string	-	등록된 modbus 신호의 이름

#### ● 알아두기

다중신호의 경우 해당 명령어로 삭제할 수 없습니다.

### 리턴

값	설명
0	성공
음수값	실패

### 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 # Modbus IO 신호가 "di1", "do1" 로 등록되어 있는데,
2 # 이 신호 등록을 삭제하고자 할 때 하기 명령을 사용합니다. .
3 del_modbus_signal("di1")      # "di1" 점점 등록 삭제
4 del_modbus_signal("do1")      # "do1" 점점 등록 삭제

```

## 9.4.6 del\_modbus\_signal\_multi(name)

### 기능

등록된 Modbus의 다중신호를 삭제합니다. Modbus I/O 설정의 경우 티치펜던트 I/O set-up 메뉴에서 설정해야 하지만 티치펜던트 사용이 어려운 경우에 테스트를 위해서만 본 명령어를 사용하시기 바랍니다. 이 명령어를 사용하여 셋팅한 경우 티치펜던트에서 Modbus 관련 메뉴가 동작하지 않습니다.

### 인수

인수명	자료형	기본값	설명
name	string	-	등록된 modbus 다중 신호의 이름

### 리턴

값	설명
0	성공
음수값	실패

### 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시

예외	설명
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 #Modbus 다중신호 "multi1"(cnt=5)가 등록되어 있는 경우
2 # 이 신호 등록을 삭제하고자 할 때 하기 명령을 사용합니다. .
3 del_modbus_signal_multi("multi1") # "multi1" 접점 등록 삭제

```

## 9.4.7 set\_modbus\_output(iobus, value)

### 기능

외부 Modbus 장치에 신호를 보내내기 위한 명령어입니다.

Function Code 05 Write Single Coil Register

Function Code 06 Write Signle Holding Register

### 인수

인수명	자료형	기본값	설명
iobus	string	-	modbus 이름(TP에서 설정)
value	int	-	Modbus Coil Register인 경우 value <ul style="list-style-type: none"> <li>• ON : 1</li> <li>• OFF : 0</li> </ul> Modbus Holding Register인 경우 value

### ● 알아두기

다중 신호로 등록된 경우 신호이름에 인덱스[n]를 붙여 사용 가능합니다.

## 리턴

값	설명
0	성공
음수값	실패

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 #Modbus Coil Register "do1", "do2"가 등록되어 있는 경우
2 set_modbus_output("do1", ON)
3 set_modbus_output("do2", OFF)
4
5 #Modbus Holding Register "reg1", "reg2"가 등록되어 있는 경우
6 set_modbus_output("reg1", 10)
7 set_modbus_output("reg2", 24)
8
9 #다중신호 신호이름: "multi"(cnt=2)가 등록되어 있는 경우
10 #“multi[0]”, “multi[1]” 사용 가능
11 set_modbus_output("multi[0]", 24)
12 set_modbus_output("multi[1]", 65535)

```

## 9.4.8 set\_modbus\_outputs(iobus\_list, val\_list)

### 기능

Modbus Slave장치에 복수 개의 신호를 내보내기 위한 명령입니다.

최대 출력 가능 개수는 32개 입니다.

### 인수

인수명	자료형	기본값	설명
iobus	string	-	modbus 이름(TP에서 설정)
value	int	-	I/O 출력 값 list

### 리턴

값	설명
0	성공
음수값	실패

### 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 # Modbus digital I/O 출력 접점 "d1"에 OFF, "d2"에 ON, "d3"에 ON
2 set_modbus_outputs(iobus_list=["d1", "d2", "d3"], val_list=[0,1,1])
3 # Modbus digital I/O 출력 접점 "d3"에 OFF, "d4"에 ON
4 set_modbus_outputs(["d3", "d4"], [0,1])

```

## 9.4.9 set\_modbus\_output\_multi(iobus, val\_list)

### 기능

외부 Modbus 장치에 신호를 보내내기 위한 명령어입니다.

Function Code 15 Write Multiple Coil Registers

Function Code 16 Write Multiple Holding Registers

### 인수

인수명	자료형	기본값	설명
iobus	string	-	modbus 다중신호 이름(TP에서 설정)
Val_list	list		Modbus 다중신호의 value list

#### ❶ 알아두기

다중 신호 이름에 등록된 신호 개수와 출력 값 list의 원소 개수가 맞지 않으면 에러가 발생합니다.

### 리턴

값	설명
0	성공
음수값	실패

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 #Modbus Coil Register가 "do1"(cnt=5), "do2"(cnt=3)로 등록되어 있는 경우
2 set_modbus_output_multi("do1", [ON, OFF, ON, ON, ON])
3 set_modbus_output_multi("do2", [ON, ON, ON])
4
5 #Modbus Holding Register가 "reg1"(cnt=5), "reg2"(cnt=3),로 등록되어 있는 경우
6 set_modbus_output_multi("reg1", [10, 101, 12345, 777, 555])
7 set_modbus_output_multi("reg2", [24, 25, 26])

```

## 9.4.10 get\_modbus\_input(iobus)

### 기능

Modbus Slave 장치에서 신호를 읽어오기 위한 명령어입니다.

### 인수

인수명	자료형	기본값	설명
iobus	string	-	modbus 이름(TP에서 설정)

#### ● 알아두기

다중신호로 등록된 경우 신호이름에 인덱스[n]를 붙여 사용 가능합니다.

## 리턴

값	설명
0 or 1	Modbus Digital I/O 인 경우 ON or OFF
value	Modbus Analog 모듈인 경우 해당 레지스터 값

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 #Modbus Coil Register "di1", "di2"가 등록되어 있는 경우
2 get_modbus_input("di1")      # return = ON
3 get_modbus_input("di2")      # return = OFF
4
5 #Modbus Holding Register "reg1", "reg2"가 등록되어 있는 경우
6 get_modbus_input("reg1")    # return = 1234
7 get_modbus_input("reg2")    # return = 5555
8
9 #다중신호 신호이름: "multi"(cnt=2)가 등록되어 있는 경우
10 #“multi[0]”, “multi[1]” 사용 가능
11 get_modbus_input("multi1[0]") # return = 10
12 get_modbus_input("multi2[1]") # return = 25

```

### 9.4.11 get\_modbus\_inputs(iobus\_list)

#### 기능

Modbus Slave장치에 복수 개의 신호를 읽어오기 위한 명령입니다.

#### 인수

인수명	자료형	기본값	설명
iobus_list	list(string)	-	modbus input 이름 list (TP에서 설정) signal type이 아래의 경우에만 사용가능 <ul style="list-style-type: none"> <li>• DR_MODBUS_DIG_INPUT</li> <li>• DR_MODBUS_DIG_OUTPUT</li> </ul>

#### 리턴

값	설명
int (>=0)	한번에 읽은 복수 개의 신호 값 (iobus_list 의 첫번째 값=LSB, iobus_list 의 마지막 값=MSB 가 되는 비트 조합의 값)
음수 값	실패

#### 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 # Modbus digital I/O 입력 신호가 "di1"은 OFF, "di2"은 ON, "di3"은 ON 인 경우
2 res = get_modbus_inputs(iobus_list=["di1", "di2", "di3"])
   #res 기대값 = 0b110(이진수), 6(십진수), 0x06(16진수)
3
4
5 # Modbus digital I/O 입력 신호가 "di4"은 OFF, "di5"은 ON 인 경우
6 res = get_modbus_inputs(["di4", "di5"])
   #res 기대값 = 0b10(이진수), 2(십진수), 0x02(16진수)
7

```

### 9.4.12 get\_modbus\_inputs\_list(iobus\_list)

#### 기능

Modbus Slave 장치의 Register Type의 복수 개 신호를 읽어오기 위한 명령어입니다.

#### 인수

인수명	자료형	기본값	설명
iobus_list	list(string)	-	modbus input 이름 list (TP에서 설정) signal type이 아래의 경우에만 사용 가능 <ul style="list-style-type: none"> <li>• DR_MODBUS_REG_INPUT</li> <li>• DR_MODBUS_REG_OUTPUT</li> </ul>

#### 리턴

값	설명
res	읽은 값의 갯수
val_list	한번에 읽은 복수 개의 신호 값의 list

#### 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시

예외	설명
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 # Modbus Register I/O 입력 신호 "Holding1"은 1234, "Input1"은 567,
2   "Holding2"은 9876 인 경우
3   res, val_list = get_modbus_inputs_list(iobus_list=["Holding1", "Input1", "H
4     olding2"])
      #res 기대값 = 3
      #val_list 기대값 = [1234, 567, 9876]

```

### 9.4.13 get\_modbus\_input\_multi (iobus)

#### 기능

Modbus Slave 장치에서 신호를 읽어오기 위한 명령어입니다.

#### 인수

인수명	자료형	기본값	설명
iobus	string	-	modbus 다중신호 이름(TP에서 설정)

#### 리턴

값	설명
res	성공여부: 성공 0, 실패 음수 값
list	해당 신호의 cnt에 해당하는 입력 값 list

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 #다중신호 신호이름: "multi"(cnt=2)가 등록되어 있는 경우
2 #“multi[0]”, “multi[1]” 사용가능
3 get_modbus_input_multi("multi1")    # return = [10, 101]

```

## 9.4.14 wait\_modbus\_input(iobus, val, timeout=None)

### 기능

Modbus Slave 장치에서 지정한 신호값이 val(ON or OFF)이 될 때까지 대기합니다. 대기 시간은 timeout 설정으로 변경할 수 있으며, 지정된 시간이 지나면 대기 상태가 종료됨과 동시에 결과를 리턴합니다. 단, timeout을 설정하지 않으면 무한 대기합니다.

### 인수

인수명	자료형	기본값	설명
iobus	string	-	modbus 이름 (TP에서 설정)
value	int	-	Modbus digital I/O 인 경우 <ul style="list-style-type: none"> <li>• ON : 1</li> <li>• OFF : 0</li> </ul>
			Modbus analog I/O 인 경우 value
timeout	float	-	대기 시간 [sec] 설정하지 않으면 무한 대기

### 알아두기

다중신호로 등록된 경우 신호이름에 인덱스[n]를 붙여 사용 가능합니다.

## 리턴

값	설명
0	성공
-1	실패 (time-out)

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1  wait_modbus_input("CIN0", ON)  #"CIN0" 신호가 ON될 때까지 무한 대기
2  wait_modbus_input("CIN0", OFF)  #"CIN0" 신호가 OFF될 때까지 무한 대기
3
4  res = wait_modbus_input("CIN0", ON, 3)  #"CIN0" 신호가 ON될 때까지 3초간 대기
5  #3초 안에 1번 신호가 ON 되면, res = 0
6  #3초 안에 1번 신호가 ON 되지 않았으면 res = -1
7
8  #다중신호 신호이름: "multi"(cnt=2)가 등록되어 있는 경우
9  #“multi[0]”, “multi[1]” 사용 가능
10 wait_modbus_input("multi [1]", ON)
11 wait_modbus_input("multi [0]", OFF)

```

### 9.4.15 set\_modbus\_slave(address, val)

#### 기능

ModbusTCP Slave의 Gerneral Purpose Register 영역에 값을 보내내기 위해 사용합니다.

#### 인수

인수명	자료형	기본값	설명
address	int	-	GPR영역의 주소값 (128~255)
val	int	-	2byte 값 (0~65535)

#### 리턴

값	설명
0	성공
음수값	실패

#### 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 set_modbus_slave(128, 0)
2 set_modbus_slave(255, 65535)

```

### 9.4.16 get\_modbus\_slave(address)

#### 기능

ModbusTCP Slave의 Gerneral Purpose Register 영역을 접근하여 값을 가져오기 위해 사용합니다.

#### 인수

인수명	자료형	기본값	설명
address	int	-	읽으려고 하는 GPR영역의 주소값 (128~255)

#### 리턴

값	설명
value	해당 레지스터 값

#### 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 value1 = get_modbus_slave(128)
2 value2 = get_modbus_slave(255)

```

### 9.4.17 modbus\_crc16(data)

#### 기능

Modbus protocol 사용 시, modbus crc16 계산에 대한 부하를 줄이기 위한 명령어입니다.

#### 인수

인수명	자료형	기본값	설명
data	byte	-	crc16계산을 위한 modbus data

#### 리턴

값	설명
crchigh	Crc16 계산 결과의 high byte
crlclow	Crc16 계산 결과의 low byte

#### 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1  data = b"\x01\x02\x03\x04\x05\x06"
2  crchigh, crclow = modbus_crc16(data)
3  #crchigh = 186(DEC), BA(HEX)
4  #crclow = 221(DEC), DD(HEX)

```

### 9.4.18 modbus\_send\_make(send\_data)

#### 기능

Modbus protocol 사용 시, send data에 대하여 modbus crc16 결과값을 포함하는 결과 data를 제공하기 위한 명령어입니다.

#### 인수

인수명	자료형	기본값	설명
send_data	byte	-	crc 계산이 필요한 전송 데이터

#### 리턴

값	설명
result_data	send data에 modbus crc16 결과값이 포함된 결과 data

#### 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 senddata = b"\x01\x02\x03\x04\x05\x06"
2 resultdata = modbus_send_make(senddata)
#resultdata = b'\x01\x02\x03\x04\x05\x06\xba\xdd'

```

### 9.4.19 modbus\_recv\_check(recv\_data)

#### 기능

Modbus protocol 사용 시, receive data에 대하여 crc16 값을 이용한 데이터 무결성 체크를 하기 위한 명령어입니다.

#### 인수

인수명	자료형	기본값	설명
recv_data	byte	-	raw modbus data

#### 리턴

값	설명
res	True/False

#### 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

#### 예제

```

1 #recvdata = b"\x01\x02\x03\x04\x05\x06\xba\xdd"
2 res = modbus_recv_check(recvdata)

```

	3      #recv = True
--	---------------------

## 9.4.20 modbus\_unsigned\_to\_signed(unsigned\_data)

### 기능

Modbus protocol 사용 시, 2bytes unsigned 데이터를 signed data로 변환하기 위한 명령어입니다.

### 인수

인수명	자료형	기본값	설명
unsigned_data	int	-	2byte unsigned data(0~65535)

### 리턴

값	설명
signed_data	2byte signed data(-32769 ~ 32767)

### 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

### 예제

	1      unsigned_data = 40000 2      signed_data = modbus_unsigned_to_signed(unsigned_data)
--	---

## 9.5 Industrial Ethernet (EtherNet/IP,PROFINET)

### 9.5.1 set\_output\_register\_bit(address, val)

#### 기능

Industrial Ethernet(EtherNet/IP, PROFINET) Slave의 Output영역의 Bit General Purpose Register의 값을 설정하기 위한 명령어입니다.

#### 인수

인수명	자료형	기본값	설명
address	unsigned short	-	Industrial Ethernet Slave의 Output Bit GPR영역 주소값 (0-63)
val	int	-	ON : 1 OFF : 0

#### 리턴

값	설명
0	성공
음수값	실패

#### 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 set_output_register_bit (0, ON)
2 set_output_register_bit (63, OFF)

```

### 9.5.2 set\_output\_register\_int(address, val)

#### 기능

Industrial Ethernet(EtherNet/IP, PROFINET) Slave의 Output영역의 Int General Purpose Register의 값을 설정하기 위한 명령어입니다.

#### 인수

인수명	자료형	기본값	설명
address	unsigned short	-	Industrial Ethernet Slave의 Output Int GPR영역 주소값 (0-23)
val	int	-	int value (4byte)

#### 리턴

값	설명
0	성공
음수값	실패

#### 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시

예외	설명
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 set_output_register_int (0, 0x00FF00FF)
2 set_output_register_int (23, 65535)

```

### 9.5.3 set\_output\_register\_float(address, val)

#### 기능

Industrial Ethernet(EtherNet/IP, PROFINET) Slave의 Output영역의 Float General Purpose Register의 값을 설정하기 위한 명령어입니다.

#### 인수

인수명	자료형	기본값	설명
address	unsigned short	-	Industrial Ethernet Slave의 Output Float GPR영역 주소값 (0-23)
val	float	-	float value (4byte)

#### 리턴

값	설명
0	성공
음수값	실패

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 set_output _register_float (0, 4.5)
2 set_output _register_float (23, 2.3)

```

## 9.5.4 get\_output\_register\_bit(address)

### 기능

Industrial Ethernet(EtherNet/IP, PROFINET) Slave의 Output영역의 Bit General Purpose Register의 값을 읽어오기 위한 명령어입니다.

### 인수

인수명	자료형	기본값	설명
address	unsigned short	-	Industrial Ethernet Slave의 Output Bit GPR영역 주소값 (0-63)

### 리턴

값	설명
value	해당 GPR 주소의 값

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 a = get_output_register_bit (0)
2 b = get_output_register_bit (63)

```

## 9.5.5 get\_output\_register\_int(address)

### 기능

Industrial Ethernet(EtherNet/IP, PROFINET) Slave의 Output영역의 Int General Purpose Register의 값을 읽어오기 위한 명령어입니다.

### 인수

인수명	자료형	기본값	설명
address	unsigned short	-	Industrial Ethernet Slave의 Output Int GPR영역 주소값 (0-23)

### 리턴

값	설명
value	해당 GPR 주소의 값

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 a = get_output_register_int (0)
2 b = get_output_register_int(23)

```

## 9.5.6 get\_output\_register\_float(address)

### 기능

Industrial Ethernet(EtherNet/IP, PROFINET) Slave의 Output영역의 Float General Purpose Register의 값을 읽어오기 위한 명령어입니다.

### 인수

인수명	자료형	기본값	설명
address	unsigned short	-	Industrial Ethernet Slave의 Output Float GPR영역 주소값 (0-23)

### 리턴

값	설명
value	해당 GPR 주소의 값

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 a = get_output_register_float (0)
2 b = get_output_register_float (63)

```

## 9.5.7 get\_input\_register\_bit(address)

### 기능

Industrial Ethernet(EtherNet/IP, PROFINET) Slave의 Input영역의 Bit General Purpose Register의 값을 읽어오기 위한 명령어입니다.

### 인수

인수명	자료형	기본값	설명
address	unsigned short	-	Industrial Ethernet Slave의 Input Bit GPR영역 주소값 (0-63)

### 리턴

값	설명
value	해당 GPR 주소의 값

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 a = get_input_register_bit (0)
2 b = get_input_register_bit (63)

```

## 9.5.8 get\_input\_register\_int(address)

### 기능

Industrial Ethernet(EtherNet/IP, PROFINET) Slave의 Input영역의 Int General Purpose Register의 값을 읽어오기 위한 명령어입니다.

### 인수

인수명	자료형	기본값	설명
address	unsigned short	-	Industrial Ethernet Slave의 Input Int GPR영역 주소값 (0-63)

### 리턴

값	설명
value	해당 GPR 주소의 값

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 a = get_input_register_int (0)
2 b = get_input_register_int (23)

```

## 9.5.9 get\_input\_register\_float(address)

### 기능

Industrial Ethernet(EtherNet/IP, PROFINET) Slave의 Input영역의 Float General Purpose Register의 값을 읽어오기 위한 명령어입니다.

### 인수

인수명	자료형	기본값	설명
address	unsigned short	-	Industrial Ethernet Slave의 Input Float GPR영역 주소값 (0-23)

### 리턴

값	설명
value	해당 GPR 주소의 값

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 a = get_input_register_float (0)
2 b = get_input_register_float (63)

```

## 9.6 FOCAS

### 9.6.1 focas\_connect(ip, port, timeout)

#### 기능

Machine Center Controller와의 연결을 하기 위해 사용되는 명령어입니다. 정상연결시 0이상의 핸들값을 반환 합니다.

연결가능한 controller는 다음과 같습니다.

- FANUC Series 30i/31i/32i/35i-MODEL B
- FANUC Series 31i-MODEL B5
- FANUC Series Power Motion i-MODEL A
- FANUC Series 0i-MODEL D/F

#### 인수

인수명	자료형	기본값	설명
ip	str	-	Server IP address: (예) "192.168.137.200"
port	int	-	Server Port number (예) 8193

인수명	자료형	기본값	설명
timeout	int	-	timeout설정(0:의 경우 무한, 단위:s)

### 리턴

값	설명
Handle	성공(핸들값 리턴)
errorCode	에러코드(focas_get_error_str 참조)

### 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

### 예제

1	ErrCode, hMachineCenter = focas_connect("10.10.0.95", 8193, 10)
---	---

## 9.6.2 focas\_disconnect(handle)

### 기능

Machine Center Controller와의 연결을 해제하기 위해 사용되는 명령어입니다.

### 인수

인수명	자료형	기본값	설명
handle	int	-	FOCAS 사용에 필요한 통신 고유의 제어상수값

## 리턴

값	설명
errorCode	0 : 성공 0이외의 값 : 에러(focas_get_error_str 참조)

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 ErrCode, hMachineCenter = focas_connect("10.10.0.95", 8193, 10)
2 ErrCode = focas_disconnect(hMachineCenter)

```

### 9.6.3 focas\_pmc\_read\_bit(handle, addr\_type, start\_num, bit\_offset)

## 기능

Machine Center Controller의 PMC Data를 읽기 위해서 사용되는 명령어입니다. 데이터의 반환 형태가 bit일 경우 사용합니다.

#### 주의

DRL구동전 해당 컨트롤러의 PMC Signal Map을 반드시 확인 후 구동하시기 바랍니다.

## 인수

인수명	자료형	기본값	설명
handle	int	-	machine_center_connect DRL을 통해 얻어온 handle value
addr_type	str		G (Output signal from PMC to CNC) F (Input signal to PMC from CNC) Y (Output signal to PMC from machine) X (Input signal from PMC to machine) A (Message display) R (Internal relay) T (Timer) K (Keep relay) C (Counter) D (Data table) M (Input signal from other PMC path) N (Output signal to other PMC path) E (Extra relay) Z (System relay) • 대소문자 구분안함
start_num	Int		Strart Address Number(0~9999)
bit_offset	int		Bit Offset(0~7)

## 리턴

값	설명
errorCode	0 : 성공(정상으로 통신해제) 0이외의 값 : 에러(focas_get_error_str 참조)
pmc_data	PMC데이터(bit형)

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 ErrCode, hMachineCenter = focas_connect("10.10.0.95", 8193, 10)
2 ErrCode, PMC_Data = focas_pmc_read_bit(hMachineCenter,"R", 3500, 0)
3 ErrCode = focas_disconnect(hMachineCenter)

```

## 9.6.4 focas\_pmc\_read\_char(handle, addr\_type, start\_num, read\_count)

### 기능

Machine Center Controller의 PMC Data를 읽기 위해서 사용되는 명령어입니다. 데이터의 반환 형태가 char(1Byte)일 경우 사용합니다.

#### ⚠ 주의

DRL 구동전 해당 컨트롤러의 PMC Signal Map을 반드시 확인 후 구동하시기 바랍니다.

### 인수

인수명	자료형	기본값	설명
handle	int	-	FOCAS 사용에 필요한 통신 고유의 제어상수값

인수명	자료형	기본값	설명
addr_type	str		G (Output signal from PMC to CNC) F (Input signal to PMC from CNC) Y (Output signal to PMC from machine) X (Input signal from PMC to machine) A (Message display) R (Internal relay) T (Timer) K (Keep relay) C (Counter) D (Data table) M (Input signal from other PMC path) N (Output signal to other PMC path) E (Extra relay) Z (System relay) • 대소문자 구분안함
start_num	Int		Start Address Number(0~9999)
read_count	int		Start Address Number에서 읽고 싶은 char의 개수(최대 5)

### 리턴

값	설명
errorCode	0 : 성공(정상으로 통신해제) 0이외의 값 : 에러(focas_get_error_str 참조)
pmc_data	PMC데이터(char형)

### 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시

예외	설명
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 ErrCode, hMachineCenter = focas_connect("10.10.0.95", 8193, 10)
2 ErrCode, PMC_Data = focas_pmc_read_char(hMachineCenter,"R",100,3)
3 ErrCode = focas_disconnect(hMachineCenter)

```

## 9.6.5 focas\_pmc\_read\_word(handle, addr\_type, start\_num, read\_count)

### 기능

Machine Center Controller의 PMC Data를 읽기 위해서 사용되는 명령어입니다.  
데이터의 반환 형태가 word(2Byte)일 경우 사용합니다.

#### 주의

DRL구동전 해당 컨트롤러의 PMC Signal Map을 반드시 확인 후 구동하시기 바랍니다.

### 인수

인수명	자료형	기본값	설명
handle	int	-	FOCAS 사용에 필요한 통신 고유의 제어상수값

인수명	자료형	기본값	설명
addr_type	str		G (Output signal from PMC to CNC) F (Input signal to PMC from CNC) Y (Output signal to PMC from machine) X (Input signal from PMC to machine) A (Message display) R (Internal relay) T (Timer) K (Keep relay) C (Counter) D (Data table) M (Input signal from other PMC path) N (Output signal to other PMC path) E (Extra relay) Z (System relay) • 대소문자 구분안함
start_num	Int		Start Address Number(0~9999)
read_count	int		Start Address Number에서 읽고 싶은 word의 개수(최대 5)

## 리턴

값	설명
errorCode	0:성공(정상으로 통신해제) 0이외의 값 : 에러(focas_get_error_str 참조)
pmc_data	PMC데이터(word형)

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시

예외	설명
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

### 예제

```

1 ErrCode, hMachineCenter = focas_connect("10.10.0.95", 8193, 10)
2 ErrCode, PMC_Data = focas_pmc_read_word(hMachineCenter,"R",3500,3)
3 ErrCode = focas_disconnect(hMachineCenter)

```

## 9.6.6 focas\_pmc\_read\_long(handle, addr\_type, start\_num, read\_count)

### 기능

Machine Center Controller의 PMC Data를 읽기 위해서 사용되는 명령어입니다.

데이터의 반환 형태가 long(4Byte)일 경우 사용합니다.

#### 주의

DRL 구동 전 해당 컨트롤러의 PMC Signal Map을 반드시 확인 후 구동하시기 바랍니다.

### 인수

인수명	자료형	기본값	설명
handle	int	-	FOCAS 사용에 필요한 통신 고유의 제어상수값

인수명	자료형	기본값	설명
addr_type	str		G (Output signal from PMC to CNC) F (Input signal to PMC from CNC) Y (Output signal to PMC from machine) X (Input signal from PMC to machine) A (Message display) R (Internal relay) T (Timer) K (Keep relay) C (Counter) D (Data table) M (Input signal from other PMC path) N (Output signal to other PMC path) E (Extra relay) Z (System relay) • 대소문자 구분안함
start_num	Int		Start Address Number(0~9999)
read_count	int		Start Address Number에서 읽고 싶은 long의 개수(최대 5)

### 리턴

값	설명
errorCode	0:성공(정상으로 통신해제) 0이외의 값 : 에러(focas_get_error_str 참조)
pmc_data	PMC데이터(long형)

### 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시

예외	설명
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

### 예제

```

1 ErrCode, hMachineCenter = focas_connect("10.10.0.95", 8193, 10)
2 ErrCode, PMC_Data = focas_pmc_read_long(hMachineCenter,"R",3500,3)
3 ErrCode = focas_disconnect(hMachineCenter)

```

## 9.6.7 focas\_pmc\_read\_float(handle, addr\_type, start\_num, read\_count)

### 기능

Machine Center Controller의 PMC Data를 읽기 위해서 사용되는 명령어입니다.

데이터의 반환 형태가 float(4Byte, 32-bit-floatring-point-type)일 경우 사용합니다.

#### ⚠ 주의

DRL 구동 전 해당 컨트롤러의 PMC Signal Map을 반드시 확인 후 구동하시기 바랍니다.

### 인수

인수명	자료형	기본값	설명
handle	int	-	FOCAS 사용에 필요한 통신 고유의 제어상수값

인수명	자료형	기본값	설명
addr_type	str		G (Output signal from PMC to CNC) F (Input signal to PMC from CNC) Y (Output signal to PMC from machine) X (Input signal from PMC to machine) A (Message display) R (Internal relay) T (Timer) K (Keep relay) C (Counter) D (Data table) M (Input signal from other PMC path) N (Output signal to other PMC path) E (Extra relay) Z (System relay) • 대소문자 구분안함
start_num	Int		Start Address Number(0~9999)
read_count	int		Start Address Number에서 읽고 싶은 float의 개수(최대 5)

### 리턴

값	설명
errorCode	0:성공(정상으로 통신해제) 0이외의 값 : 에러(focas_get_error_str 참조)
pmc_data	PMC데이터(float형)

### 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시

예외	설명
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 ErrCode, hMachineCenter = focas_connect("10.10.0.95", 8193, 10)
2 ErrCode, PMC_Data = focas_pmc_read_float(hMachineCenter, "D", 10, 3)
3 ErrCode = focas_disconnect(hMachineCenter)

```

## 9.6.8 focas\_pmc\_read\_double(handle, addr\_type, start\_num, read\_count)

### 기능

Machine Center Controller의 PMC Data를 읽기 위해서 사용되는 명령어입니다.

데이터의 반환 형태가 double(8Byte, 64-bit-floatring-point-type)일 경우 사용합니다.

#### 주의

DRL 구동 전 해당 컨트롤러의 PMC Signal Map을 반드시 확인 후 구동하시기 바랍니다.

### 인수

인수명	자료형	기본값	설명
handle	int	-	FOCAS 사용에 필요한 통신 고유의 제어상수값

인수명	자료형	기본값	설명
addr_type	str		G (Output signal from PMC to CNC) F (Input signal to PMC from CNC) Y (Output signal to PMC from machine) X (Input signal from PMC to machine) A (Message display) R (Internal relay) T (Timer) K (Keep relay) C (Counter) D (Data table) M (Input signal from other PMC path) N (Output signal to other PMC path) E (Extra relay) Z (System relay) • 대소문자 구분안함
start_num	Int		Start Address Number(0~9999)
read_count	int		Start Address Number에서 읽고 싶은 double의 개수(최대 5)

### 리턴

값	설명
errorCode	0:성공(정상으로 통신해제) 0이외의 값 : 에러(focas_get_error_str 참조)
pmc_data	PMC데이터(double형)

### 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시

예외	설명
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

### 예제

```

1 ErrCode, hMachineCenter = focas_connect("10.10.0.95", 8193, 10)
2 ErrCode, PMC_Data = focas_pmc_read_double(hMachineCenter, "D", 10, 3)
3 ErrCode = focas_disconnect(hMachineCenter)

```

## 9.6.9 focas\_pmc\_write\_bit(handle, addr\_type, start\_num, bit\_offset, write\_data)

### 기능

Machine Center Controller의 PMC Data를 쓰기 위해서 사용되는 명령어입니다.  
데이터의 반환 형태가 bit일 경우 사용합니다.

#### ⚠ 주의

DRL 구동 전 해당 컨트롤러의 PMC Signal Map을 반드시 확인 후 구동하시기 바랍니다.

### 인수

인수명	자료형	기본값	설명
handle	int	-	FOCAS 사용에 필요한 통신 고유의 제어상수값

인수명	자료형	기본값	설명
addr_type	str		G (Output signal from PMC to CNC) F (Input signal to PMC from CNC) Y (Output signal to PMC from machine) X (Input signal from PMC to machine) A (Message display) R (Internal relay) T (Timer) K (Keep relay) C (Counter) D (Data table) M (Input signal from other PMC path) N (Output signal to other PMC path) E (Extra relay) Z (System relay) • 대소문자 구분안함
start_num	Int		Strart Address Number(0~9999)
bit_offset	int		Bit Offset(0~7)
write_data	Int		ON : 1 OFF : 0

### 리턴

값	설명
errorCode	0:성공(정상으로 통신해제) 0이외의 값 : 에러(focas_get_error_str 참조)

### 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시

예외	설명
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 ErrCode, hMachineCenter = focas_connect("10.10.0.95", 8193, 10)
2 ErrCode = focas_pmc_write_bit(hMachineCenter, "R", 3000, 0, ON)
3 ErrCode = focas_disconnect(hMachineCenter)

```

### 9.6.10 focas\_pmc\_write\_char(handle, addr\_type, start\_num, write\_data, write\_count)

## 기능

Machine Center Controller의 PMC Data를 쓰기 위해서 사용되는 명령어입니다.  
데이터의 반환 형태가 char(1Byte)일 경우 사용합니다.

### 주의

DRL 구동 전 해당 컨트롤러의 PMC Signal Map을 반드시 확인 후 구동하시기 바랍니다.

## 인수

인수명	자료형	기본값	설명
handle	int	-	FOCAS 사용에 필요한 통신 고유의 제어상수값

인수명	자료형	기본값	설명
addr_type	str		G (Output signal from PMC to CNC) F (Input signal to PMC from CNC) Y (Output signal to PMC from machine) X (Input signal from PMC to machine) A (Message display) R (Internal relay) T (Timer) K (Keep relay) C (Counter) D (Data table) M (Input signal from other PMC path) N (Output signal to other PMC path) E (Extra relay) Z (System relay) • 대소문자 구분안함
start_num	Int		Strart Address Number(0~9999)
write_data	byte		전송할 데이터의 형태는 byte입니다.
write_count	Int		전송할 char Data의 개수를 입력합니다. 최대 5

## 리턴

값	설명
errorCode	0:성공(정상으로 통신해제) 0이외의 값 : 에러(focas_get_error_str 참조)

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시

예외	설명
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 ErrCode, hMachineCenter = focas_connect("10.10.0.95", 8193, 10)
2 MC_Command = [129, 213]
3 ErrCode = focas_pmc_write_char(hMachineCenter, "R", 3000, MC_Command, 2)
4 ErrCode = focas_disconnect(hMachineCenter)

```

### 9.6.11 focas\_pmc\_write\_word(handle, addr\_type, start\_num, write\_data, write\_count)

## 기능

Machine Center Controller의 PMC Data를 쓰기 위해서 사용되는 명령어입니다.

데이터의 반환 형태가 word(2Byte)일 경우 사용합니다.

#### ⚠ 주의

DRL 구동 전 해당 컨트롤러의 PMC Signal Map을 반드시 확인 후 구동하시기 바랍니다.

## 인수

인수명	자료형	기본값	설명
handle	int	-	FOCAS 사용에 필요한 통신 고유의 제어상수값

인수명	자료형	기본값	설명
addr_type	str		G (Output signal from PMC to CNC) F (Input signal to PMC from CNC) Y (Output signal to PMC from machine) X (Input signal from PMC to machine) A (Message display) R (Internal relay) T (Timer) K (Keep relay) C (Counter) D (Data table) M (Input signal from other PMC path) N (Output signal to other PMC path) E (Extra relay) Z (System relay) • 대소문자 구분안함
start_num	Int		Start Address Number(0~9999)
write_data	word		전송할 데이터의 형태는 word(2Byte)입니다.
write_count	Int		전송할 word Data의 개수를 입력합니다. 최대 5

## 리턴

값	설명
errorCode	0:성공(정상으로 통신해제) 0이외의 값 : 에러(focas_get_error_str 참조)

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시

예외	설명
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 ErrCode, hMachineCenter = focas_connect("10.10.0.95", 8193, 10)
2 MC_Command = [-129, 1213]
3 ErrCode = focas_pmc_write_word(hMachineCenter, "R", 3100, MC_Command, 2)
4 ErrCode = focas_disconnect(hMachineCenter)

```

### 9.6.12 focas\_pmc\_write\_long(handle, addr\_type, start\_num, write\_data, write\_count)

## 기능

Machine Center Controller의 PMC Data를 쓰기 위해서 사용되는 명령어입니다.

데이터의 반환 형태가 long(4Byte)일 경우 사용합니다.

#### ⚠ 주의

DRL 구동 전 해당 컨트롤러의 PMC Signal Map을 반드시 확인 후 구동하시기 바랍니다.

## 인수

인수명	자료형	기본값	설명
handle	int	-	FOCAS 사용에 필요한 통신 고유의 제어상수값

인수명	자료형	기본값	설명
addr_type	str		G (Output signal from PMC to CNC) F (Input signal to PMC from CNC) Y (Output signal to PMC from machine) X (Input signal from PMC to machine) A (Message display) R (Internal relay) T (Timer) K (Keep relay) C (Counter) D (Data table) M (Input signal from other PMC path) N (Output signal to other PMC path) E (Extra relay) Z (System relay) • 대소문자 구분안함
start_num	Int		Strart Address Number(0~9999)
write_data	int		전송할 데이터의 형태는 long(4Byte)입니다.
write_count	Int		전송할 long Data의 개수를 입력합니다. 최대 5

## 리턴

값	설명
errorCode	0:성공(정상으로 통신해제) 0이외의 값 : 에러(focas_get_error_str 참조)

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시

예외	설명
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 ErrCode, hMachineCenter = focas_connect("10.10.0.95", 8193, 10)
2 MC_Command = [-7129, 11213]
3 ErrCode = focas_pmc_write_long(hMachineCenter, "G", 3100, MC_Command, 2)
4 ErrCode = focas_disconnect(hMachineCenter)

```

### 9.6.13 focas\_pmc\_write\_float(handle, addr\_type, start\_num, write\_data, write\_count)

## 기능

Machine Center Controller의 PMC Data를 쓰기 위해서 사용되는 명령어입니다.

데이터의 반환 형태가 float(4Byte, 32-bit-floatring-point-type)일 경우 사용합니다.

#### ⚠ 주의

DRL 구동 전 해당 컨트롤러의 PMC Signal Map을 반드시 확인 후 구동하시기 바랍니다.

## 인수

인수명	자료형	기본값	설명
handle	int	-	FOCAS 사용에 필요한 통신 고유의 제어상수값

인수명	자료형	기본값	설명
addr_type	str		G (Output signal from PMC to CNC) F (Input signal to PMC from CNC) Y (Output signal to PMC from machine) X (Input signal from PMC to machine) A (Message display) R (Internal relay) T (Timer) K (Keep relay) C (Counter) D (Data table) M (Input signal from other PMC path) N (Output signal to other PMC path) E (Extra relay) Z (System relay) • 대소문자 구분안함
start_num	Int		Strart Address Number(0~9999)
write_data	float		전송할 데이터의 형태는 float(4Byte)입니다.
write_count	Int		전송할 float Data의 개수를 입력합니다. 최대 5

## 리턴

값	설명
errorCode	0 : 성공(정상으로 통신해제) 0이외의 값 : 에러(focas_get_error_str 참조)

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시

예외	설명
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 ErrCode, hMachineCenter = focas_connect("10.10.0.95", 8193, 10)
2 MC_Command = [-178.12, 11.478]
3 ErrCode = focas_pmc_write_float(hMachineCenter, "G", 3100, MC_Command, 2)
4 ErrCode = focas_disconnect(hMachineCenter)

```

### 9.6.14 focas\_pmc\_write\_double(handle, addr\_type, start\_num, write\_data, write\_count)

## 기능

Machine Center Controller의 PMC Data를 쓰기 위해서 사용되는 명령어입니다.

데이터의 반환 형태가 double(8Byte, 64-bit-floatring-point-type)일 경우 사용합니다.

#### ⚠ 주의

DRL 구동 전 해당 컨트롤러의 PMC Signal Map을 반드시 확인 후 구동하시기 바랍니다.

## 인수

인수명	자료형	기본값	설명
handle	int	-	FOCAS 사용에 필요한 통신 고유의 제어상수값

인수명	자료형	기본값	설명
addr_type	str		G (Output signal from PMC to CNC) F (Input signal to PMC from CNC) Y (Output signal to PMC from machine) X (Input signal from PMC to machine) A (Message display) R (Internal relay) T (Timer) K (Keep relay) C (Counter) D (Data table) M (Input signal from other PMC path) N (Output signal to other PMC path) E (Extra relay) Z (System relay) • 대소문자 구분안함
start_num	Int		Strart Address Number(0~9999)
write_data	double		전송할 데이터의 형태는 double(8Byte)입니다.
write_count	Int		전송할 double Data의 개수를 입력합니다. 최대 5

## 리턴

값	설명
errorCode	0 : 성공(정상으로 통신해제) 0이외의 값 : 에러(focas_get_error_str 참조)

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시

예외	설명
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 ErrCode, hMachineCenter = focas_connect("10.10.0.95", 8193, 10)
2 MC_Command = [-178.12, 11.478]
3 ErrCode = focas_pmc_write_double(hMachineCenter, "G", 3100, MC_Command, 2)
4 ErrCode = focas_disconnect(hMachineCenter)

```

## 9.6.15 focas\_get\_error\_str(handle, errorCode)

### 기능

Focas Library관련 기능 사용시 반환되는 errorCode의 분석을 위해서 사용합니다. 에러코드 입력시 관련 에러의 원인에 대해서 string으로 반환합니다.

### 인수

인수명	자료형	기본값	설명
handle	int	-	FOCAS 사용에 필요한 통신 고유의 제어상수값
errorCode	int	-	FOCAS관련 DRL 수행후 반환되는 에러코드

### 리턴

값	설명
errorCodeString	에러코드의 상세 내역(string)

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 상세 예러코드

예외	설명
.-17	<p>상세 원인에 따라 아래의 메시지가 출력됨</p> <p>"The send data is larger than the maximum transfer unit"</p> <p>"The sending data size is illegal"</p> <p>"The number of the received packet is 0"</p> <p>"The mark of the protocol is incorrect in the received packet header"</p> <p>"The packet type flag is incorrect in the received packet header"</p> <p>"The flag of the direction is incorrect in the received packet header"</p> <p>"Illegal received data size"</p> <p>"Communication error in the Ethernet Board "</p> <p>"protocol error"</p>

예외	설명
-16	상세 원인에 따라 아래의 메시지가 출력됨 "Error of socket API function " "Error of connect API function" "Error of send API function" "Error of recv API function" "Error of select API function" "Error of setsockopt API function" "Error of gethostbyname API function" "Timeout error of send API function" "Timeout error of recv API function" "Error of Winsock API is occurred in other process" "EOF (end of file) detected " "socket error"
-15	"DLL not exist error"
-14	"error in APi library initial valiefie"
-13	"low temperature alarm of intelligent terminal"
-12	"hight temperature alarm of intelligent terminal"
-11	"bus error"
-10	"system error"
-9	"hssb communication error"
-8	"library handle error"
-7	"CNC/PMC version missmatch"
-6	"abnormal error"
-5	"system error"
-4	"shared RAM parity error"

예외	설명
-3	"emm386 or mmcbsys install error"
-2	"reset or stop occurred error"
-1	"busy error"
0	"no problem"
1	"command prepare error or pmc not exist"
2	"data block length error"
3	"data number error or address range error"
4	"data attribute error or data type error"
5	"data error"
6	"no option error"
7	"write protect error"
8	"memory overflow error"
9	"cnc parameter not correct error"
10	"buffer error"
11	"path error"
12	"cnc mode error"
13	"execution rejected error"
14	"data server error"
15	"alarm has been occurred"
16	"CNC is not running"

예외	설명
17	"protection data error"
18	"error generated by PMC"
19	"PMC handle error"
20	"overwrite stop in program read"
21	"reset interrupt in program read"
-100	"Library opening failed."
-101	"The maximum number of machine tools that can be connected has been exceeded."
-102	"The handle is not connected"

### 예제 상세 예러코드

```

1 ErrCode, hMachineCenter = focas_connect("10.10.0.95", 8193, 10)
2 if ErrCode != 0 :
3     ErrorString = focas_get_error_str(hMachineCenter, ErrCode)
4     MC_Command = [-178.12, 11.478]
5     focas_pmc_write_double(hMachineCenter, "G", 3100, MC_Command, 2)
6     focas_disconnect(hMachineCenter)

```

## 10 외부 비전 명령어

### 10.1 개요

임의의 외부 비전 시스템을 당사 로봇과 연결하여 작업(Vision guided robotics) 하기 위한 명령어를 제공합니다. 평면 상에 위치하는 물체의 2차원 변위 정보(위치Tx, Ty, 회전Rz)를 측정하는 2D비전 시스템에 연결이 가능하며, 다수 물체에 대한 측정정보 입력도 가능하도록 명령어가 구성되어 있습니다. 2D 비전 시스템을 사용하는 비전-가이드 로봇의 경우 작업대상 물체에 대하여 정의된 로봇작업 좌표를 기준 이미지와 측정이미지 간의 비전측정좌표 변위 정보를 이용하여 로봇작업 좌표를 산출합니다. 2D비전 시스템을 이용한 로봇 애플리케이션의 설치 및 작업은 비전시스템의 이미지 좌표계를 로봇시스템의 물리적 좌표계로 캘리브레이션(좌표계 보정) 시키는 것이 필요하며, 외부 비전시스템을 사용할 때에는 비전 자체적으로 좌표계 캘리브레이션 작업을 수행한 후 보정된 좌표정보를 로봇에 전송해야 합니다.

비전시스템의 설치는 로봇과 연결된 Eye-in-hand방식과 로봇과 분리된 in-line 방식으로 모두 설치할 수 있으며, 작업하는 상황에서 로봇과 비전시스템의 위치관계가 변하지 않도록 고정되어 있어야 합니다. 비전시스템과 로봇제어기는 TCP/IP방식으로 통신하며, 로봇제어기의 유선공유기 포트에 비전시스템 케이블을 연결해주어야 통신이 가능합니다.

### 10.2 2D Vision - COGNEX / SICK / VISOR

2D Vision(COGNEX, SICK, VISOR) 명령어는 다음과 같이 구성되어 있습니다.

No.	구분	명령어
1	제조사 선택	<a href="#">vs_set_info(type)(p. 441)</a>
2	카메라 연결	<a href="#">vs_connect(ip_addr, port_num=9999)(p. 442)</a> <a href="#">vs_disconnect()(p. 443)</a>
3	비전 작업 관리	<a href="#">vs_get_job()(p. 443)</a> <a href="#">vs_set_job(job_name)(p. 444)</a> <a href="#">cognex_set_integer (job_name, integer_number)(p. 445)</a> <a href="#">visor_job_change(index)(p. 445)</a>
4	사물 인식 및 감지	<a href="#">vs_trigger()(p. 446)</a>
5	로봇 태스크	<a href="#">vs_set_init_pos(vision_posx_init, robot_posx_init, vs_pos=1)(p. 447)</a>

No.	구분	명령어
		<a href="#">vs_get_offset_pos(vision_posx_meas, vs_pos=1)(p. 449)</a>
6	사용자 정의	<a href="#">vs_request(cmd)(p. 451)</a>
		<a href="#">vs_result()(p. 451)</a>

### 10.2.1 vs\_set\_info(type)

#### 기능

사용할 비전시스템의 종류를 설정한다.

#### 인수

인수명	자료형	기본값	설명
type	int	DR_VS_CUSTOM(0) DR_VS_COGNEX(1) DR_VS_SICK(2) DR_VS_VISOR(3)	

#### 리턴

값	설명
type의 ID	설정하고자 하는 type의 ID

#### 예제

```

1  vs_set_info(DR_VS_COGNEX) #Vision type information setting
2  vs_connect("192.168.137.10") #Connect to vision - Vision IP, Default port
3  # Enter your task
4  vs_disconnect()           #Disconnect to vision

```

- 지원 모델

Type	Model
DR_VS_COGNEX	COGNEX IS2000M-23M Series COGNEX IS5600/5705 Series COGNEX IS7000Series COGNEX IS8000Series
DR_VS_SICK	SICK Inspector PIM60 Series SICK Inspector PI50 Series
DR_VS_VISOR	Sensopart V20 Series
DR_VS_CUSTOM	

### 10.2.2 vs\_connect(ip\_addr, port\_num=9999)

#### 기능

비전시스템의 통신을 연결한다.

#### 인수

인수명	자료형	기본값	설명
ip_addr	str	-	비전 모듈의 Server IP (예, 192.168.137.200)
port_num	int	9999	Port 번호 (예, 9999)

#### 리턴

값	설명
0	연결 성공
-1	연결 실패

## 예제

```

1  vs_set_info(DR_VS_COGNEX) #Vision type information setting
2  vs_connect("192.168.137.10") #Connect to vision - Vision IP, Default port
3  # Enter your task
4  vs_disconnect()           #Disconnect to vision

```

### 10.2.3 vs\_disconnect()

#### 기능

비전시스템의 통신을 해제한다.

#### 리턴

값	설명
없음	

## 예제

```

1  vs_set_info(DR_VS_COGNEX) #Vision type information setting
2  vs_connect("192.168.137.10") #Connect to vision - Vision IP, Default port
3  # Enter your task
4  vs_disconnect()           #Disconnect to vision

```

### 10.2.4 vs\_get\_job()

#### 기능

비전시스템에 현재 로딩되어 있는 작업명을 불러온다. (\*VS\_TYPE: DR\_VS\_COGNEX, DR\_VS\_SICK)

#### 리턴

값	자료형	설명
job_name	string	연결 성공
-1	int	연결 실패

## 예제

```

1 vs_set_info(DR_VS_COGNEX) #Vision type information setting
2 vs_connect("192.168.137.10") #Connect to vision - Vision IP, Default port
3
4 vs_set_job("test.job")      # Set (load) the current vision job
5 job_name=vs_get_job()      # Get the current setting vision job
6 tp_popup("{0}".format(job_name))
7
8 vs_disconnect()            # Disconnect to vision

```

### 10.2.5 vs\_set\_job(job\_name)

#### 기능

입력된 작업을 비전시스템에 로딩한다. (\*VS\_TYPE: DR\_VS\_COGNEX, DR\_VS\_SICK)

#### 인수

인수명	자료형	기본값	설명
job_name	string		로딩할 작업명

#### 리턴

값	자료형	설명
0	int	성공
-1	int	실패

## 예제

```

1 vs_set_info(DR_VS_COGNEX) #Vision type information setting
2 vs_connect("192.168.137.10") #Connect to vision - Vision IP, Default port
3
4 vs_set_job("test.job")      # Set (load) the current vision job
5 job_name=vs_get_job()      # Get the current setting vision job
6 tp_popup("{0}".format(job_name))
7
8 vs_disconnect()            # Disconnect to vision

```

## 10.2.6 cognex\_set\_integer (job\_name, integer\_number)

### 기능

해당 작업의 입력된 index에 로딩 합니다.

(\*VS\_TYPE: DR\_VS\_COGNEX)

### 리턴

값	설명
1	성공
-1	실패

### 예제

```

1   vs_set_info(DR_VS_COGNEX)           #Vision type information setting
2   vs_connect("192.168.137.10")        #Connect to vision - Vision IP, Default
3   port
4   res = cognex_set_integer("Pattern_1.Scale_Tolerance",1)    # Enter your
5   task
6   tp_popup("{0}".format(res))         #Display message
7   vs_disconnect()                   #Disconnect to vision

```

## 10.2.7 visor\_job\_change(index)

### 기능

입력된 번호로 비전센서의 설정작업으로 로딩합니다.

(\*VS\_TYPE: DR\_VS\_VISOR)

### 인수

인수명	자료형	기본값	설명
index	int	-	변경하고자 하는 Job number

## 리턴

값	설명
result	잡 변경 성공 (1) 잡 변경 실패 (-1)

## 예제

```

1 vs_set_info(DR_VS_VISOR)           # Select type of vision sensor
2 vs_connect("192.168.137.101")      # Vision IP, Default port
3 vis_posx = posx (410,310,300,0,0,0) # Define the initial posx data -
vision
4 rob_posx = posx (400,300,300,0,180,0) # Define the initial posx data -
robot
5 vs_set_init_pos(vis_posx, rob_posx, VS_POS1) # Enter the initial posx data
to Vision
6 visor_job_change(2)                 # change the job as the input
parameter
7
8 for i in range(10):
9     pos, var_list = vs_trigger()       # Execute the vision
measurment
10    tp_popup("{0}".format(var_list))
11
12    if var_list[0] == 1:               # Check the inspection result
13        robot_posx_meas = vs_get_offset_pos(pos, VS_POS1) # offset the
robot pose
14        tp_popup("{0}".format(robot_posx_meas))
15    else:
16        tp_popup("Inspection Fail")
17 vs_disconnect()

```

## 10.2.8 vs\_trigger()

### 기능

비전시스템에 측정명령을 전달한다. 측정이 성공하면, 측정결과값을 리턴한다.

(VS\_TYPE: DR\_VS\_COGNEX, DR\_VS\_SICK)

## 리턴

값	자료형	설명
pos	posx	pos: 대상물체의 측정 위치 (posx 변수 타입)
var_list	list[float]	var_list: 사용자가 추가로 입력한 측정결과값 예) 검사합불, 거리측정, 각도측정값 등
-1, []	int	실패

## 예제

```

1  vs_set_info(DR_VS_COGNEX)           # Select type of vision sensor
2  vs_connect("192.168.137.10")        # Vision IP, Default port
3
4  vis_posx = posx (410,310,300,0,0,0)      # Define the initial posx
data - vision
5  rob_posx = posx (400,300,300,0,180,0)    # Define the initial posx data -
robot
6
7  vs_set_init_pos(vis_posx, rob_posx, VS_POS1) # Enter the initial posx data
to Vision
8
9  for i in range(10):
10     pos, var_list = vs_trigger()          # Execute the vision
meausrement
11     if var_list[0] == 1:                  # Check the inspection result
12         robot_posx_meas = vs_get_offset_pos(pos, VS_POS1) # offset the
robot pose
13         moveL(robot_posx_meas)           # move the robot pose
14     else:
15         tp_popup("Inspection Fail")
16
17 vs_disconnect()

```

## 10.2.9 vs\_set\_init\_pos(vision\_posx\_init, robot\_posx\_init, vs\_pos=1)

### 기능

비전 가이던스 작업을 수행할 대상물의 초기 위치정보를 입력한다.  
(\*VS\_TYPE: DR\_VS\_COGNEX, DR\_VS\_SICK)

## 인수

인수명	자료형	기본값	설명
vision_posx_init	posx	-	비전측정 좌표 초기값
robot_posx_init	posx	-	로봇작업 좌표 초기값
vs_pos	int	1	입력한 초기값의 pos 번호

## 리턴

값	자료형	설명
vs_pos	int	성공 - 입력된 pos 번호
-1	int	실패

## 예제

```

1 vs_set_info(DR_VS_COGNEX)           # Select type of vision sensor
2 vs_connect("192.168.137.10")        # Vision IP, Default port
3 vis_posx = posx (410,310,300,0,0,0) # Define the initial posx
data - vision
4 rob_posx = posx (400,300,300,0,180,0) # Define the initial posx data -
robot
5 vs_set_init_pos(vis_posx, rob_posx, VS_POS1) # Enter the initial posx data
to Vision
6 for i in range(10):
7     pos, var_list = vs_trigger()          # Execute the vision
measurment
8     if var_list[0] == 1:                  # Check the inspection result
9         robot_posx_meas = vs_get_offset_pos(pos, VS_POS1) # offset the
robot pose
10    moveL(robot_posx_meas)             # move the robot pose
11 else:
12     tp_popup("Inspection Fail")
13 vs_disconnect()

```

## 10.2.10 vs\_get\_offset\_pos(vision\_posx\_meas, vs\_pos=1)

### 기능

비전 시스템에서 측정된 좌표값을 이용하여 옵셋(가이드)된 로봇작업 좌표를 산출한다.

사전에 vs\_set\_init\_pos를 통해 초기값이 입력되어 있어야 한다.

### 인수

인수명	자료형	기본값	설명
vision_posx_meas	posx	-	vs_trigger를 이용하여 얻어진 비전측정 좌표값
vs_pos	int	1	옵셋 좌표를 계산할 로봇 초기값 pos 번호

### 리턴

값	자료형	설명
robot_posx_meas	posx	성공
-1	int	실패

### 예제

```

1  vs_set_info(DR_VS_COGNEX)           # Select type of vision sensor
2  vs_connect("192.168.137.10")        # Vision IP, Default port
3
4  vis_posx = posx (410,310,300,0,0,0)    # Define the initial posx
data - vision
5  rob_posx = posx (400,300,300,0,180,0)   # Define the initial posx data -
robot
6
7  vs_set_init_pos(vis_posx, rob_posx, VS_POS1) # Enter the initial posx data
to Vision
8
9  for i in range(10):
10     pos, var_list = vs_trigger()          # Execute the vision
meausrement
11     if var_list[0] == 1:                  # Check the inspection result
12         robot_posx_meas = vs_get_offset_pos(pos, VS_POS1) # offset the
robot pose
13         moveL(robot_posx_meas)           # move the robot pose
14     else:

```

```

15     tp_popup("Inspection Fail")
16
17     vs_disconnect()

```

### 10.2.11 통합예제 - DR\_VS\_COGNEX, DR\_VS\_SICK

#### 예제

```

1  vs_set_info(DR_VS_COGNEX)                      # Select type of vision
2  sensor
3  if ( vs_connect("192.168.137.10") != 0 ):      # Vision IP, Default
4    port
5    tp_popup("connection fail",DR_PM_MESSAGE)
6    exit()
7
8  vis_posx_init = posx (410,310,300,0,0,0)        # Define the initial
9  posx data - vision
10 rob_posx_init1 = posx (400,300,300,0,180,0)      # Define the initial posx
11 data - robot
12 rob_posx_init2 = posx (420,320,300,0,180,0)      # Define the initial
13 posx data - robot
14
15 vs_set_init_pos(vis_posx_init, rob_posx_init1, VS_POS1) # Enter the
16 initial posx data to Vision
17 vs_set_init_pos(vis_posx_init, rob_posx_init2, VS_POS2)
18
19 for i in range(10):
20   pos_meas, var_list = vs_trigger()      # Execute the vision measusrement
21   if pos_meas==1:                      # Vision Fail to measure the object
22     tp_popup("Vision measure fail")
23     continue
24   if var_list[0] == 1:                  # Check the inspection result
25     # Get guided posx data
26     rob_posx1_meas = vs_get_offset_pos(pos_meas, VS_POS1) # offset the
27     robot pose
28     rob_posx2_meas = vs_get_offset_pos(pos_meas, VS_POS2) # offset the
29     robot pose
30     movel(rob_posx1_meas)
31     movel(rob_posx2_meas)
32   else:
33     tp_popup("Inspection Fail")
34     continue
35
36 vs_disconnect()

```

## 10.2.12 vs\_request(cmd)

### 기능

Vision System에 요청할 기능을 설정합니다.

(\*VS\_TYPE: DR\_VS\_CUSTOM)

### 인수

인수명	자료형	기본값	설명
cmd	int	-	비전시스템에서 검출하고자 하는 물체 개수

### 리턴

값	설명
0	성공
-1	실패
-2	통신 타임 아웃(3초) 발생

### 예제

1	<pre>vs_request(1)          # request the vision measurement on the "1" job</pre>
---	---

## 10.2.13 vs\_result()

### 기능

Vision System의 처리 결과를 가져옵니다.

(\*VS\_TYPE: DR\_VS\_CUSTOM)

## 리턴

값	설명
cnt (>=1)	성공 비전시스템에서 검출된 물체 개수
result	비전 결과 위치 리스트 (x좌표, y좌표, 회전값)
-	실패 시, cnt == -2 res = 빈 리스트

## 예제

```

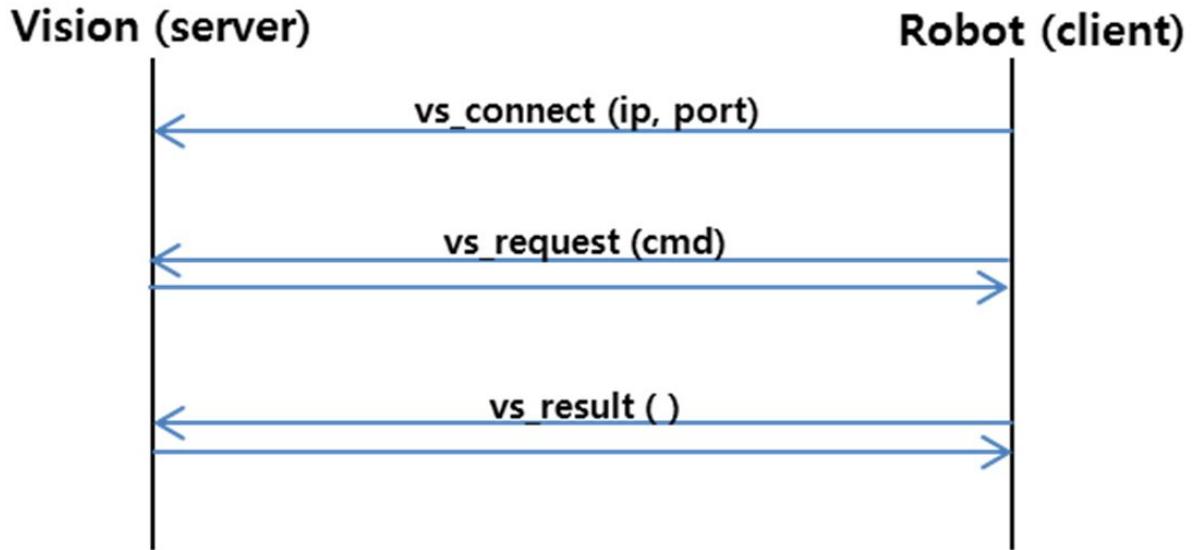
1 vs_set_info(DR_VS_CUSTOM)
2 res = vs_connect("192.168.137.200", 9999)           #Vision and communication
connection attempt
3 if res !=0:          #Check the result of communication connection
    tp_popup("connection fail",DR_PM_MESSAGE)  #If connection fails, program
ends
4     exit()
5
6 ret = vs_request(1)      #Request for object vision measurement information
7
8
9 cnt, result = vs_result()          # Get object measurement result
information
10
11 for i in range(cnt):
12     x = result[i][0]
13     y = result[i][1]
14     t = result[i][2]
15     tp_popup("x={0},y={1}, t={2}".format(result[i][0], result[i][1],
result[i][2]),DR_PM_MESSAGE)

```

## 10.2.14 통합예제 - DR\_VS\_CUSTOM

### 통신프로토콜

비전 시스템에서 하기 프로토콜을 준수해야만, 비전 명령어가 정상적으로 동작합니다.



### vs\_request (cmd)

1. 로봇 제어기 → 비전 시스템
  - “MEAS\_START” +cmd[4byte]
    - cmd는 검출 물체 개수: integer를 4byte로 변환 ex) cmd=1 → 00000001
    - ex) cmd=1 인 경우: “MEAS\_START”+00000001
      - 실제 패킷: 4D4541535F535441525400000001
  - 2. 비전 시스템 → 로봇 제어기
    - 비전 시스템이 정상인 경우 “MEAS\_OK” 비 정상인 경우 “MEAS\_NG”를 전송

### vs\_result ( )

1. 로봇 제어기 → 비전 시스템
  - “MEAS\_REQUEST”
2. 비전 시스템 → 로봇 제어기
  - “MEAS\_INFO” +cnt[4byte] +[(x[4byte] + y[4byte] + t[4byte]) x cnt개수]
    - cnt는 검출된 물체의 개수를 의미 함.
    - x(x좌표), y(y좌표), t(회전값)은 100배 스케일링한 값을 전송해야 함.
    - ex) cnt = 1, (x=1.1, y=2.2, t=3.3)
      - “MEAS\_INFO”+1[4byte] +110[4byte] +220[4byte] +330[4byte]
      - 실제 패킷: 4D4541535F494E464F0000000100000006E000000DC0000014A
    - ex) cnt = 2, (x=1.1, y=2.2, t=3.3) (x=1.1, y=-2.2, t=-3.3)
      - “MEAS\_INFO”+2[4byte] +110[4byte] +220[4byte] +330[4byte] +110[4byte] -220[4byte] -330[4byte]
      - 실제 패킷:
 

4D4541535F494E464F0000000200000006E000000DC0000014A00000006EFFFFF24FF  
FFFEB6

## 예제

```

1 vs_set_info(DR_VS_CUSTOM)
2 res = vs_connect("192.168.137.200", 9999)           #비전과 통신 연결 시도
3 if res !=0:                                         #통신 연결 결과 확인
4     tp_popup("connection fail",DR_PM_MESSAGE)      #접속 실패 시, 프로그램 종료
5     exit()
6
7 ret = vs_request(1)                                #1번 물체 비전측정 정보 요청
8
9 cnt, result = vs_result()                          #물체 측정결과 정보 가져오기
10
11 for i in range(cnt):
12     x = result[i][0]
13     y = result[i][1]
14     t = result[i][2]
15     tp_popup("x={0},y={1}, t={2}".format(result[i][0], result[i][1],
result[i][2]),DR_PM_MESSAGE)

```

## 10.3 Pickit 3D

Pickit 3D 명령어는 다음과 같이 구성되어 있습니다.

No.	구분	명령어
1	카메라 연결	pickit_connect(ip)(p. 455)
		pickit_disconnect()(p. 455)
2	비전 작업 관리	pickit_change_configuration(setup_id, product_id))(p. 456)
3	카메라 보정	pickit_request_calibration()(p. 456)
4	사물 인식 및 감지	pickit_detection(offset_z)(p. 457)
		pickit_next_object(offset_z)(p. 459)
5	백업	pickit_save_snapshot()(p. 460)

### 10.3.1 pickit\_connect(ip)

#### 기능

비전시스템의 통신을 연결한다.

PickIt의 Default IP는 192.168.66.1로 사용자가 PickIt IP를 Robot IP와 동일 대역대로 맞춰서 사용한다.

#### 인수

인수명	자료형	기본값	설명
ip_addr	str	-	Pickit 3D의 Server IP (예, 192.168.137.90)

#### 리턴

값	설명
0	연결 성공
-1	연결 실패

#### 예제

```

1   pickit_connect("192.168.137.90") #Connect to vision - Vision IP
2   pickit_disconnect()               #Disconnect to vision

```

### 10.3.2 pickit\_disconnect()

#### 기능

비전시스템의 통신을 해제한다.

#### 리턴

값	설명
없음	

## 예제

```

1 pickit_connect("192.168.137.90")#Connect to vision - Vision IP
2 pickit_disconnect()           #Disconnect to vision

```

### 10.3.3 pickit\_change\_configuration(setup\_id, product\_id))

#### 기능

비전시스템에 설정한 setup\_id, product\_id를 불러온다.

#### 인수

인수명	자료형	기본값	설명
Setup_id	int	-	Pickit 서버에 저장한 setup_id 번호를 입력한다.
Product_id	int		Pickit 서버에 저장한 product_id 번호를 입력한다.

#### 리턴

값	설명
0	연결 성공
-1	연결 실패

## 예제

```

1 pickit_connect("192.168.137.90")
2 pickit_change_configuration(6, 8) # These numbers are defined in Pickit
3 pickit_disconnect()

```

### 10.3.4 pickit\_request\_calibration()

#### 기능

비전시스템에 켈리브레이션을 1회 요청한다.

## 리턴

값	설명
0	연결 성공
-1	연결 실패

## 예제

```

1 pickit_connect("192.168.137.90")
2 pickit_request_calibration()
3 pickit_disconnect()
```

### 10.3.5 pickit\_detection(offset\_z)

#### 기능

입력된 모델을 검출하고 pick\_prepos, pick\_pos를 반환한다.

#### 인수

인수명	자료형	기본값	설명
offset_z	int	-	'pick_prepos' 거리를 설정한다.

## 리턴

값	자료형	설명
<pre>data_dictionary =  {'pick_pos':pick_pos, 'pick_prepos':pick_prepos, 'object_age':data['object_age'], 'object_type':data['object_type'], 'object_dimensions':data['object_dimensions'], 'object_remaining':data['objects_remaining'], 'status':data['status']</pre>	<pre>{'pick_pos': posx, 'pick_prepos': posx, 'object_age': int, 'object_type': int, 'object_dimensions': int, 'object_remaining': int, 'status': int}</pre>	<p><b>'pick_pos'</b>: Model recognition position,  <b>'pick_prepos'</b>: Offset Value of model recognition position,  <b>'object_age'</b>: The amount of time that has passed between the capturing of the camera data and the moment the object information is sent to the robot. This value has to be divided by the <b>MULT</b> factor.,  <b>'object_type'</b>: The type of object detected at object_pose ,  <b>'object_dimensions'</b>: When reading array elements, each value has to be divided by the <b>MULT</b> factor.,  <b>'object_remaining'</b>: Only one object per pickit_to_robot_data message can be communicated. If this field is non-zero, it contains the number of remaining objects that can be sent in next messages to the robot.,  <b>'status'</b>: Contains the Pickit status or a response to previously received robot commands.</p>

## 예제

```

1 set_singular_handling(DR_AVOID)
2 set_velj(60.0)
3 set_accj(100.0)
4 set_velx(250.0, 80.625)
5 set_accx(1000.0, 322.5)
6
7 pickit_connect("192.168.137.90")
8 pickit_change_configuration( 7,10 ) # These numbers are defined in Pickit
9
10 data = pickit_detection(100)
11 if data['status'] == ResponseStatus.OBJECTS_FOUND:
12     # Picking motion
13     moveL(data['pick_prepos'])
14     moveL(data['pick_pos'])
15     moveL(data['pick_prepos'])
16
17 pickit_disconnect()
```

### 10.3.6 pickit\_next\_object(offset\_z)

#### 기능

입력된 모델의 다음으로 검출된 pick\_prepos, pick\_pos를 반환한다.

#### 인수

인수명	자료형	기본값	설명
offset_z	int	-	'pick_prepos' 거리를 설정한다.

#### 리턴

값	자료형	설명
<pre>data_dictionary =  {'pick_pos':pick_pos, 'pick_prepos':pick_prepos, 'object_age':data['object_age'], 'object_type':data['object_type'], 'object_dimensions':data['object_dimensions'], 'object_remaining':data['objects_remaining'], 'status':data['status']}</pre>	<pre>{'pick_pos': posx, 'pick_prepos': posx, 'object_age': int, 'object_type': int, 'object_dimensions': int, 'object_remaining': int, 'status': int}</pre>	<p><b>'pick_pos'</b>: 모델 인식 위치,  <b>'pick_prepos'</b>: 모델 인식 위치의 오프셋 값,  <b>'object_age'</b>: The amount of time that has passed between the capturing of the camera data and the moment the object information is sent to the robot. This value has to be divided by the <b>MULT</b> factor.,  <b>'object_type'</b>: The type of object detected at object_pose  <b>'object_dimensions'</b>: When reading array elements, each value has to be divided by the <b>MULT</b> factor.,  <b>'object_remaining'</b>: Only one object per pickit_to_robot_data message can be communicated. If this field is non-zero, it contains the number of remaining objects that can be sent in next messages to the robot.,  <b>'status'</b>: Contains the Pickit status or a response to previously received robot commands.</p>

#### 예제

```

1 pickit_connect("192.168.137.90")
2 pickit_change_configuration(7, 10) # These numbers are defined in Pickit
3
4 set_singular_handling(DR_AVOID)
5 set_velj(60.0)
6 set_accj(100.0)

```

```

7   set_velx(250.0, 80.625)
8   set_accx(1000.0, 322.5)
9   home_pos = posx(122.09, 35.28, 303.63, 140.45, 158.9, 134.39)
10  while True:
11      # Detection
12      data = pickit_detection(100)
13      if data['status'] == ResponseStatus.OBJECTS_FOUND:
14          # Picking motion
15          movel(data['pick_prepos'])
16          movel(data['pick_pos'])
17          movel(data['pick_prepos'])

18          remaining = data['object_remaining']
19          while remaining > 0:
20              data = pickit_next_object(100)
21              remaining = data['object_remaining']

22              # Picking motion
23              movel(data['pick_prepos'])
24              movel(data['pick_pos'])
25              movel(data['pick_prepos'])

26      pickit_disconnect()

```

### 10.3.7 pickit\_save\_snapshot()

#### 기능

snapshot을 서버에 저장한다.

#### 리턴

값	설명
0	연결 성공
-1	연결 실패

#### 예제

1	picket_connect("192.168.137.90")
2	picket_save_snapshot()
3	picket_disconnect()

### 10.3.8 통합 예제 - Pickit 3D

#### 예제 1

```

1  ##### For robot-camera calibration example #####
2
3  set_singular_handling(DR_AVOID)
4  set_velj(60.0)
5  set_accj(100.0)
6  set_velx(250.0, 80.625)
7  set_accx(1000.0, 322.5)
8  pickit_connect("192.168.137.90")
9
10 # Move to Pose #
11 pos1= posx(476.76, -151.68, 384.83, 33.36, 24.71, 95.44)
12 pos2= posx(495.86, -150.08, 435.69, 1.43, 8.21, 122.25)
13 pos3= posx(508.79, -83.06, 446.94, 82.88, -35.91, 39.23)
14 pos4= posx(521.66, -146.28, 431.8, 29.71, -33.73, 82.42)
15 pos5= posx(508.35, -147.45, 386.37, 101.03, 38.04, 42.68)
16 movel(pos1)
17 pickit_request_calibration()
18 movel(pos2)
19 pickit_request_calibration()
20 movel(pos3)
21 pickit_request_calibration()
22 movel(pos4)
23 pickit_request_calibration()
24 movel(pos5)
25 pickit_request_calibration()

```

#### 예제 2

```

1  ##### For simple picking example #####
2  pickit_connect("192.168.137.90")
3  pickit_change_configuration( 7,10 ) # These numbers are defined in Pickit
4
5  set_singular_handling(DR_AVOID)
6  set_velj(60.0)
7  set_accj(100.0)
8  set_velx(250.0, 80.625)
9  set_accx(1000.0, 322.5)
10 home_pos = posx(122.09, 35.28, 303.63, 140.45, 158.9, 134.39)
11     #Move to home pose
12 movel(home_pos)
13
14     # Detection
15 data = pickit_detection(100)
16 if data['status'] == ResponseStatus.OBJECTS_FOUND:

```

```

17     # Picking motion
18     movel(data['pick_prepos'])
19     movel(data['pick_pos'])
20     movel(data['pick_prepos'])
21
22 pickit_disconnect()

```

## 예제 3

```

1     ### For multiple parts picking example ####
2
3     pickit_connect("192.168.137.90")
4     pickit_change_configuration(7, 10) # These numbers are defined in Pickit
5
6     set_singular_handling(DR_AVOID)
7     set_velj(60.0)
8     set_accj(100.0)
9     set_velx(250.0, 80.625)
10    set_accx(1000.0, 322.5)
11    home_pos = posx(122.09, 35.28, 303.63, 140.45, 158.9, 134.39)
12    while True:
13        # Move to home pose
14        movel(home_pos)
15
16        # Detection
17        data = pickit_detection(100)
18        if data['status'] == ResponseStatus.OBJECTS_FOUND:
19
20            # Picking motion
21            movel(data['pick_prepos'])
22            movel(data['pick_pos'])
23            movel(data['pick_prepos'])
24
25            remaining = data['object_remaining']
26            while remaining > 0:
27                data = pickit_next_object(100)
28                remaining = data['object_remaining']
29
30            # Picking motion
31            movel(data['pick_prepos'])
32            movel(data['pick_pos'])
33            movel(data['pick_prepos'])

```

## 11 두산비전(SVM) 명령어

두산비전 모듈(SVM) 명령어는 다음과 같이 구성되어 있습니다.

구분	세부 사항	명령어
1 카메라 연결		(3.2.1-ko_KR) <code>svm_connect(ip="192.168.137.5", port=20)</code> (p. 464)  <code>svm_disconnect()</code> (p. 465)
2 이미지 품질 체크		(3.2.1_temp-ko_KR) <code>svm_set_led_brightness(value)</code> <sup>26</sup>  <code>svm_get_led_brightness()</code> (p. 466)  (3.2.1_temp-ko_KR) <code>svm_set_camera_exp_val(value)</code> <sup>27</sup>  <code>svm_set_camera_gain_val(value)</code> (p. 467)  (3.2.1_temp-ko_KR) <code>svm_set_camera_load(job_id)</code> <sup>28</sup>
3 비전 작업 관리		<code>svm_set_job(job_id)</code> (p. 468)
4 카메라 보정		<code>svm_get_robot_pose(job_id)</code> (p. 469)
5 사물 인식 및 감지	사물 인식	<code>svm_get_vision_info(job_id)</code> (p. 470)  <code>svm_get_variable(tool_id, var_type)</code> (p. 471)
	랜드마크 인식	<code>svm_detect_landmark(job_id)</code> (p. 472)  (3.2.1_temp-ko_KR) <code>svm_get_marker_offset_pose(cpos, offset, euler_mode)</code> (p. 473)
	바코드 인식	<code>svm_get_detect_barcode()</code> (p. 474)

<sup>26</sup>[https://doosanrobotics-manual.atlassian.net/wiki/pages/createpage.action?fromPageId=202047505&linkCreation=true&spaceKey=ProgrammingMT&title=%283.2.1\\_temp-ko\\_KR%29+svm\\_set\\_led\\_brightness%28value%29](https://doosanrobotics-manual.atlassian.net/wiki/pages/createpage.action?fromPageId=202047505&linkCreation=true&spaceKey=ProgrammingMT&title=%283.2.1_temp-ko_KR%29+svm_set_led_brightness%28value%29)

<sup>27</sup>[https://doosanrobotics-manual.atlassian.net/wiki/pages/createpage.action?fromPageId=202047505&linkCreation=true&spaceKey=ProgrammingMT&title=%283.2.1\\_temp-ko\\_KR%29+svm\\_set\\_camera\\_exp\\_val%28value%29](https://doosanrobotics-manual.atlassian.net/wiki/pages/createpage.action?fromPageId=202047505&linkCreation=true&spaceKey=ProgrammingMT&title=%283.2.1_temp-ko_KR%29+svm_set_camera_exp_val%28value%29)

<sup>28</sup>[https://doosanrobotics-manual.atlassian.net/wiki/pages/createpage.action?fromPageId=202047505&linkCreation=true&spaceKey=ProgrammingMT&title=%283.2.1\\_temp-ko\\_KR%29+svm\\_set\\_camera\\_load%28job\\_id%29](https://doosanrobotics-manual.atlassian.net/wiki/pages/createpage.action?fromPageId=202047505&linkCreation=true&spaceKey=ProgrammingMT&title=%283.2.1_temp-ko_KR%29+svm_set_camera_load%28job_id%29)

구분	세부 사항	명령어
		<a href="#">svm_get_barcode_db_data(index)(p. 475)</a>
		<a href="#">svm_compare_barcode_db_data(dbdata)(p. 475)</a>
6 로봇 태스크		<a href="#">svm_set_init_pos_data(Id_list, Pos_list)(p. 476)</a>
		<a href="#">svm_get_offset_pos(posx_robot_init, job_id, tool_id)(p. 477)</a>
7 기타		<a href="#">svm_set_tp_popup(svm_flag)(p. 479)</a>

## 11.1 카메라 연결

### 11.1.1 svm\_connect(ip="192.168.137.5", port=20)

#### 기능

SVM의 통신을 연결한다.

#### 인수

인수명	자료형	기본값	설명
ip	str	"192.168.1 37.5"	SVM Server IP address
port	int	20	Port 번호

#### 리턴

값	설명
0	연결 성공
-1	연결 실패

#### 예제

```
1    svm_connect()      #Connect to vision - Default IP address and port number
```

	<pre> 2 # Enter the vision task 3 svm_disconnect() #Disconnect to vision </pre>
--	---

### 11.1.2 svm\_disconnect()

#### 기능

SVM의 통신을 해제한다.

#### 리턴

값	설명
없음	

#### 예제

	<pre> 1 svm_connect() #Connect to vision - Default IP address and port 2 # Enter the vision task 3 svm_disconnect() #Disconnect to vision </pre>
--	--

## 11.2 이미지 품질 체크

### 11.2.1 svm\_set\_led\_brightness(value)

#### 기능

SVM의 LED 밝기 값을 설정합니다.

#### 인수

인수명	자료형	기본값	설명
value	int	-	LED 밝기 값 (0-1000)

#### 리턴

값	설명
-1	실패 - 측정데이터 없음 또는 입력 변수 오류

## 예제

1	svm_connect()	# Connect to vision
2	svm_set_led_brightness(500)	# Disconnect to vision
3	svm_disconnect()	

## 11.2.2 svm\_get\_led\_brightness()

### 기능

SVM에 설정된 LED 밝기 값을 불러옵니다.

### 리턴

값	설명
int	SVM LED 밝기 값 (0-1000)
-1	실패 - 측정데이터 없음 또는 입력 변수 오류

## 예제

1	svm_connect()	# Connect to vision
2	svm_get_led_brightness()	# Disconnect to vision
3	svm_disconnect()	

## 11.2.3 svm\_set\_camera\_exp\_val(value)

### 기능

SVM의 노출 값을 설정합니다.

### 인수

인수명	자료형	기본값	설명
value	int	-	SVM 노출 값 (2,660,000 – 29,260,000)

## 리턴

값	설명
-1	실패 - 측정데이터 없음 또는 입력 변수 오류

## 예제

```

1 svm_connect()                      # Connect to vision
2 svm_set_camera_exp_val(2,660,000)
3 svm_disconnect()                   # Disconnect to vision

```

## 11.2.4 svm\_set\_camera\_gain\_val(value)

### 기능

SVM 게인 값을 설정합니다.

### 인수

인수명	자료형	기본값	설명
value	int	1	SVM 게인 값(0-1600).

## 리턴

값	설명
-1	실패 - 측정데이터 없음 또는 입력 변수 오류

## 예제

```

1 svm_connect()                      # Connect to vision
2 svm_set_camera_gain_val(500)
3 svm_disconnect()                   # Disconnect to vision

```

### 11.2.5 svm\_set\_camera\_load(job\_id)

#### 기능

SVM 해당 Job에 저장되어있는 LED밝기/노출(exp)/gain/초점 정보를 불러옵니다.

#### 인수

인수명	자료형	기본값	설명
job	int		job id(ex - 1000, 2000, 3000)

#### 리턴

값	설명
-1	실패 - 측정데이터 없음 또는 입력 변수 오류

#### 예제

```

1  svm_connect()                                # Connect to vision
2  svm_set_camera_load(job_id)
3  svm_disconnect()                            # Disconnect to vision

```

## 11.3 비전 작업 관리

### 11.3.1 svm\_set\_job(job\_id)

#### 기능

입력된 id에 해당하는 비전작업을 SVM에 로딩한다.

#### 인수

인수명	자료형	기본값	설명
job_id	int	-	비전작업 id (예. 1000, 2000, ...)

**리턴**

값	설명
0	작업 로딩 성공
-1	작업 로딩 실패

**예제**

```

1 svm_connect()           #Connect to vision - Vision IP, Default port
2 vision_test=1000        # Define vision job ID
3 svm_set_job(vision_test) # Load the vision_test (1000)
4 svm_disconnect()        #Disconnect to vision

```

## 11.4 카메라 보정

### 11.4.1 svm\_get\_robot\_pose(job\_id)

**기능**

입력된 비전작업에 설정되어 있는 로봇자세정보(조인트 좌표계)를 불러온다. 로봇 자세정보는 비전작업 수행을 위한 shoot\_pose로 활용된다.

**인수**

인수명	자료형	기본값	설명
job_id	int	-	비전작업 id (예. 1000, 2000, ...)

**리턴**

값	설명
float [6]	로봇의 조인트 좌표 정보 (posj type)
-1	실패

## 예제

```

1  svm_connect()                      # Connect to vision
2  vision_test=1000                   # Define vision job ID
3  svm_set_job(vision_test)          # Load the vision_test (1000)
4  shoot_pos=svm_get_robot_pose (vision_test) # Load the robot pose of
   vision_test
5  tp_popup("{0}".format(shoot_pos))    # Disconnect to vision
6  svm_disconnect()

```

## 11.5 사물 인식 및 감지

### 11.5.1 svm\_get\_vision\_info(job\_id)

#### 기능

입력된 비전작업에 해당하는 측정명령을 수행한다. WCM(워크셀 매니저)를 통하여 비전작업의 측정명령 상세정보를 미리 입력하여야 한다.

#### 인수

인수명	자료형	기본값	설명
job_id	int	-	비전작업 id (예. 1000, 2000, ...)

#### 리턴

값	설명
1	측정 성공 - 1개의 물체를 검출/측정에 성공했다는 의미
0	측정 실패 - 해당하는 비전작업 물체를 검출하지 못함.
-1	측정 실패 - 통신 오류 발생 (timeout)

#### 예제

```

1  svm_connect()                      # Connect to vision
2  vision_test=1000                   # Define vision job ID
3  svm_set_job(vision_test)          # Load the vision_test (1000)

```

```

4 shoot_pos=svm_get_robot_pose(vision_test) # Load the robot pose of
vision_test
5 count=svm_get_vision_info(vision_test)      # Execute the vision
measurement
6 tp_popup("{0}".format(count))              # Check the result
7 svm_disconnect()                          # Disconnect to vision

```

## 11.5.2 svm\_get\_variable(tool\_id, var\_type)

### 기능

`svm_get_vision_info`를 수행하여 물체검출/측정에 성공(1)한 경우, 검출/측정 데이터를 불러온다. 불러온 데이터에 대한 툴(tool) id와 데이터 형식 (variable type)을 입력한다.

- Position tool: POSX\_TYPE (물체위치), VALUE\_TYPE (검출 유사도)
- Presence tool: INSP\_TYPE (유무 검사결과), VALUE\_TYPE (픽셀 카운트)
- Distance tool: INSP\_TYPE (거리측정 검사결과), VALUE\_TYPE (거리측정값)
- Angle tool: INSP\_TYPE (각도측정 검사결과), VALUE\_TYPE (각도측정값)
- Diameter tool: INSP\_TYPE (직경측정 검사결과), VALUE\_TYPE (직경측정값), POSX\_TYPE (원 중심 위치)

### 인수

인수명	자료형	기본값	설명
tool_id	int	-	비전툴 id (예. 1000, 1001, 1002, ...)
var_type	int	-	POSX_TYPE: 좌표정보 변수 (posx) INSP_TYPE: 검사결과 변수 (int) VALUE_TYPE: 측정결과 변수 (int 또는 float)

### 리턴

값	설명
variable	POSX_TYPE - 좌표정보 변수, 예. Posx(x,y,z,rx,ry,rz) INSP_TYPE: 검사결과 변수 int (성공이면 1 리턴) VALUE_TYPE: 측정결과 변수 (int 또는 float)
-1	실패 - 측정데이터 없음 또는 입력 변수 오류

## 예제

```

1  svm_connect()                                # Connect to vision
2  vision_test=1000                            # Define vision job ID
3  print_insp=1001                            # Define inspection tool ID
4  box_size=1002                            # Define measurement tool ID
5  count=svm_get_vision_info(vision_test)      # Execute the vision
6  measurement
7  if (count==1):                           # Check the result
8    # Get the position information (posx) of vision_test tool
9    pos_result=svm_get_variable(vision_test, POSX_TYPE)
10   tp_popup("{0}".format(pos_result))
11
12   # Get the inspection information (PASS or Fail) of print_insp tool
13   inspection_result=svm_get_variable(print_insp, INSP_TYPE)
14   tp_popup("{0}".format(inspection_result))
15
16   # Get the distance information (distance) of box_size tool
17   measurement_result= svm_get_variable(box_size, VALUE_TYPE)
18   tp_popup("{0}".format(measurement_result))
19
vs_disconnect()                                # Disconnect to vision

```

### 11.5.3 svm\_detect\_landmark(job\_id)

#### 기능

job\_id를 이용해 landmark 정보를 카메라 좌표 기준의 값으로 검출합니다.

#### 인수

인수명	자료형	기본값	설명
job	int		job id(ex - 1000, 2000, 3000)

#### 리턴

값	자료형	설명
count	int	검출된 landmark의 개수
cpos	list[Tx,Ty,Tz,Rx,Ry,Rz]	카메라 좌표 기준의 Landmark pose
0, []	Int, list	검출 실패 시

## 예제

```

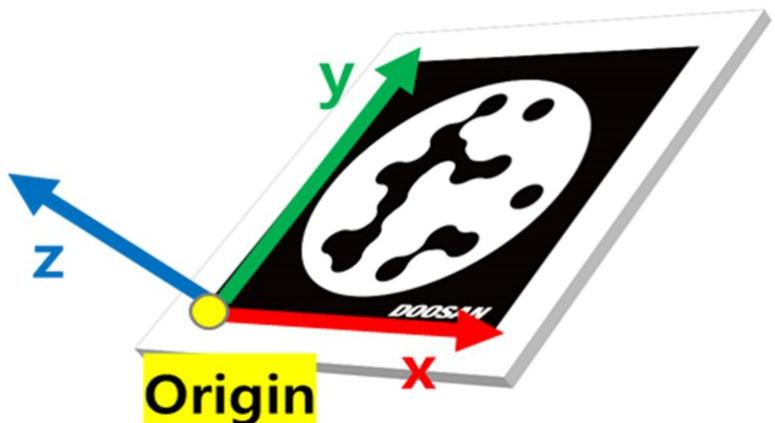
1 svm_connect()                                # Connect to vision
2 count, cpos = svm_detect_landmark(1000)
3 tp_popup("Landmark number={0}, Landmark with respect to Camera={1}").format(
4   count, cpos)
    svm_disconnect()                           # Disconnect to vision

```

### 11.5.4 svm\_get\_marker\_offset\_pose(cpos, offset, euler\_mode)

#### 기능

Landmark 원점으로부터 offset 만큼 떨어진 위치와 회전각도를 로봇 좌표 기준으로 연산합니다.



#### 인수

인수명	자료형	기본값	설명
cpos	list[Tx, Ty, Tz, Rx, Ry, Rz]		카메라 좌표 기준의 Landmark pose
offset	list[Tx, Ty, Rz]		Landmark origin으로부터 떨어진 위치와 회전각도
euler_mode	boolean		연산될 로봇좌표 기준의 Landmark pose의 orientation에 대해 “True”일때는 ZYZ, “False”일때는 XYZ

## 리턴

값	자료형	설명
rpos	list[Tx, Ty, Tz, Rx, Ry, Rz] 또는 list[Tx, Ty, Tz, Rx, Ry, Rz]	로봇좌표 기준의 Landmark pose 또는 offset이 설정된 Landmark pose
-1	Int	실패 - 측정데이터 없음 또는 입력 변수 오류

## 예제

```

1 svm_connect()                                     # Connect to vision
2
3 offset = [10,-20, 45]
4 euler_mode = True
5 rpos = svm_get_marker_offset_pose(cpos, offset, euler_mode)
6 tp_popup("Landmark with respect to Robot={0}".format(rpos))
7 svm_disconnect()                                # Disconnect to vision

```

## 11.5.5 svm\_get\_detect\_barcode()

### 기능

화면에 보이는 바코드와 QR 코드를 검출합니다.

## 리턴

값	자료형	설명
ret	int	성공 시 1 - Barcode 검출 여부
Btype	string	검출된 Barcode의 타입
bdata	string	검출된 Barcode에 담겨진 정보
0,[],[]	Int	실패 - 측정데이터 없음 또는 입력 변수 오류

## 예제

```

1 svm_connect()                                     # Connect to vision

```

```

2   ret, btype, bdata = svm_detect_barcode()
3   tp_popup("Detection={0}, Type={1}, Data={2}".format(ret,btype,bdata))
4   svm_disconnect()                                # Disconnect to vision

```

### 11.5.6 svm\_get\_barcode\_db\_data(index)

#### 기능

SVM 내 Barcode DB에 있는 값을 index를 이용하여 반환합니다.

#### 인수

인수명	자료형	기본값	설명
index	int		index(ex - 1, 2, 3, ...)

#### 리턴

값	자료형	설명
db_data	string	Barcode DB에 있는 값을 index를 이용하여 반환
“-1”	string	잘못된 index를 사용하거나, DB내 정보가 없는 경우

#### 예제

```

1   svm_connect()                                # Connect to vision
2   db_data = svm_get_barcode_db_data(1)
3   tp_popup("DB Data={0}".format(db_data))
4   svm_disconnect()                            # Disconnect to vision

```

### 11.5.7 svm\_compare\_barcode\_db\_data(dbdata)

#### 기능

입력된 인수를 SVM에 저장된 barcode DB와 비교 후 저장된 값인지 확인합니다.

## 인수

인수명	자료형	기본값	설명
dbdata	string		비교에 사용할 Barcode에 담겨진 정보

## 리턴

값	자료형	설명
0 or 1	int	비교 후 SVM 내 Barcode DB에 저장된 값인지 확인 '0'일때는 DB에 저장된 값이 아님 '1'일때는 DB에 저장된 값

## 예제

```

1 svm_connect()                                     # Connect to vision
2 ret = svm_compare_barcode_db_data(bdata)
3 tp_popup("Comparison result={0}".format(ret))
4 svm_disconnect()                                 # Disconnect to vision

```

## 11.6 로봇 태스크

### 11.6.1 svm\_set\_init\_pos\_data(Id\_list, Pos\_list)

#### 기능

비전 가이던스 작업을 수행할 대상물의 초기 id\_list와 posx\_list 정보를 입력한다.

#### 주의

- 함수 svm\_get\_offset\_pos (posx\_robot\_init, job\_id, tool\_id)를 호출하기 전에 세팅 반드시 세팅해 줄 것.
- 주의 : id\_list와 pos\_list는 각 id에 해당하는 posx와 짹과 맞춰서 사용할 것.

## 인수

인수명	자료형	기본값	설명
Id_list	List(int)	-	Id의 list ([id, id, id, ...])

인수명	자료형	기본값	설명
Pos_list	List(Posx)	-	Posx의 list (예.[posx, posx, posx, ...])

## 리턴

값	설명
없음	-

## 예제

```

1 svm_connect()                                # Connect to vision
2 vision_test=1000                            # Define vision job ID
3 count=svm_get_vision_info(vision_test)      # Execute the vision
measurement
4 if (count == 1):                           # Check the result
5     # Get the position information (posx) of vision_test tool
6     pos_result=svm_get_variable(vision_test, POSX_TYPE)
7     tp_popup("{0}".format(pos_result))
8     # Get the vision guided robot pose
9     ld_list =[vision_test]
10    pos_list =[pos_result]
11    svm_set_init_pos_data(Id_list,pos_list)
12    rob_posx=svm_get_offset_pos(posx(200,200,100,0,180,0), vision_test)
13    tp_popup("{0}".format(rob_posx))
14    # move to the rob_posx
15    movel(rob_posx, vel=30, acc=100)
16    svm_disconnect()                         # Disconnect to vision

```

## 11.6.2 svm\_get\_offset\_pos(posx\_robot\_init, job\_id, tool\_id)

### 기능

사용자가 입력한 로봇 작업 좌표정보에 비전 측정결과를 반영한 로봇 작업 좌표정보를 불러온다.

- 순서: posx\_robot\_init 입력 → vision 측정 → svm\_get\_offset\_pos 호출 → 변경된 로봇작업 좌표 (posx\_robot\_offset) 출력

## 인수

인수명	자료형	기본값	설명
posx_robot_init	posx	-	로봇 작업 좌표정보 (직접교시 등의 방법으로 입력됨)
job_id	Int	-	비전잡 id (예. 1000, 2000, 3000, ...)
tool_id	int	-	비전툴 id (예. 1000, 1001, 1002, ...)

## 리턴

값	설명
posx	비전 측정결과를 반영한 로봇 작업 좌표정보
-1	실패 - 측정데이터 없음 또는 입력 변수 오류

## 예제

```

1  svm_connect()                               # Connect to vision
2  vision_test=1000                           # Define vision job ID
3  count=svm_get_vision_info(vision_test)      # Execute the vision
4  measurement
5  if (count == 1):                          # Check the result
6      # Get the position information (posx) of vision_test tool
7      pos_result=svm_get_variable(vision_test, POSX_TYPE)
8      tp_popup("{0}".format(pos_result))
9      # Get the vision guided robot pose
10     ld_list =[vision_test]
11     pos_list =[pos_result]
12     svm_set_init_pos_data(Id_list,pos_list)
13
14     rob_posx=svm_get_offset_pos(posx(200,200,100,0,180,0), vision_test)
15     tp_popup("{0}".format(rob_posx))
16     # move to the rob_posx
17     movel(rob_posx, vel=30, acc=100)
18     svm_disconnect()                         # Disconnect to vision

```

## 11.7 기타 SVM 명령어

### 11.7.1 svm\_set\_tp\_popup(svm\_flag)

#### 기능

SVM 오류 발생 시 tp\_popup을 띄울지 말지를 설정합니다.

#### 인수

인수명	자료형	기본값	설명
svm_flag	int	1	tp_popup은 1(활성화), 0(비활성화)

#### 리턴

값	설명
없음	-

#### 예제

1 2 3	<pre>svm_set_tp_popup(<b>0</b>) svm_connect() svm_disconnect()</pre>	<pre># Hide tp_popup # Connect to vision # Disconnect to vision</pre>
-------------	--	---

## 11.8 통합예제 - SVM

### 11.8.1 예제

Vision 작업 설정 상태

- WCM에 저장된 모든 작업을 삭제한 후 아래와 같이 비전작업/툴을 생성한다.
- 비전작업생성: vision\_test (position tool, 1000)
- 비전 툴 추가: print\_insp (presence tool, 1001), box\_size (distance tool, 1002)
- TW에서 Vision Command에 “vision\_test” 작업을 선택하고, 변수정보를 설정한다.
- Custom code에 아래 예제를 추가하여 테스트를 진행한다.

1 2	<pre>svm_connect() vision_test=<b>1000</b></pre>	<pre># Connect to vision # Define vision job ID</pre>
--------	--	---

```

3   print_insp=1001                      # Define inspection tool ID
4   box_size=1002                         # Define measurement tool ID
5   svm_set_job(vision_test)              # Load the vision_test (1000)
6   movej(svm_get_robot_pose(vision_test), vel=10, acc=20) # Move to shoot
pose (movej)
7
8   if (svm_get_vision_info(vision_test)== 1): # Execute the vision
measurement
9     # Load the vision variables
10    # Get the position information (posx) of vision_test tool
11    pos_result=svm_get_variable(vision_test, POSX_TYPE)
12    tp_popup("pos_result {0}".format(pos_result))
13
14    # Get the inspection information (PASS or Fail) of print_insp tool
15    inspection_result=svm_get_variable(print_insp, INSP_TYPE)
16    tp_popup("inspection_result {0}".format(inspection_result))
17
18    # Get the distance information (distance) of box_size tool
19    measurement_result= svm_get_variable(box_size, VALUE_TYPE)
20    tp_popup("measurement_result {0}".format(measurement_result))
21
22    # Move to the vision guided robot pose
23    # Get the vision guided robot pose
24    ld_list =[vision_test]
25    pos_list =[pos_result]
26    svm_set_init_pos_data(Id_list,pos_list)
27
28    rob_posx=svm_get_offset_pos(posx(200,200,100,0,180,0), vision_test)
29    tp_popup("rob_posx {0}".format(rob_posx))
30
31    # move to the rob_posx
32    movej(rob_posx, vel=30, acc=100)
33
34  svm_disconnect()                      # Disconnect to vision

```

## 12 Application 명령어

### 12.1 External Encoder 설정 명령어

#### 12.1.1 set\_extenc\_polarity(channel, polarity\_A, polarity\_B, polarity\_Z, polarity\_S)

##### 기능

해당 엔코더 채널의 A, B상 극성과 S, Z상의 트리거 방식을 설정한다

##### 인수

인수명	자료형	기본값	설명
channel	int	1	엔코더 채널(1,2) 1: 채널 1 2: 채널 2
polarity_A	int	0	A상의 극성(0: A상, 1: /A상)
polarity_B	int	0	B상의 극성(0: B상, 1: /B상)
polarity_Z	int	0	Z상의 트리거 방식(0: 하강에지, 1: 상승에지)
polarity_S	int	0	S상의 트리거 방식(0: 하강에지, 1: 상승에지)

##### 리턴

값	설명
N/A	사용안함

##### 예외

예외	설명
N/A	사용안함

## 예제

```

1 set_extenc_polarity(1, 0, 1, 0, 1)
2 # External Encoder channel 1을 A상, /B상, Z상(하강에지), S상(상승에지)로 설정

```

## 관련 명령어

### set\_extenc\_mode

- [set\\_extenc\\_mode\(channel, mode\\_AB, pulse\\_AZ, mode\\_Z, mode\\_S, inverse\\_cnt\)\(p. 482\)](#)

## 12.1.2 set\_extenc\_mode(channel, mode\_AB, pulse\_AZ, mode\_Z, mode\_S, inverse\_cnt)

### 기능

해당 엔코더 채널의 A, B, Z, S상의 동작 모드를 설정한다.

<sup>1)</sup>V2.7.0 이전 버전 대비 mode\_S 변수 옵션 통합 - 1: Strobe Signal → Encoder Count(Clear Encoder Count Clear 단일 옵션으로 컨베이어 연동 가능), 2: Encoder Count Clear로 동작 (기존 버전 script 호환성 보장)

### 인수

인수명	자료형	기본값	설명
channel	int	1	엔코더 채널(1, 2) 1: 채널 1 2: 채널 2
mode_AB	int	0	AB상의 사용 Mode(0 ~ 4) 0: 미사용 1: A상 Quadrature 사용 B상 Quadrature 사용 2: A상 Count B상 Direction 사용 3: A상 Up Count 사용 B상 미사용 4: A상 Down Count 사용 B상 미사용

인수명	자료형	기본값	설명
pulse_AZ	int	0	Z펄스당 A펄스의 카운트 (0 ~ 100000)
mode_Z	int	0	Z상의 사용 Mode(0 ~ 1) 0: 미사용 1: A/B Count 오차 보전 2: Encoder Count Clear
mode_S	int	0	S상의 사용 Mode(0 ~ 1) 0: 미사용 1: Encoder Count Clear
inverse_cnt	int	0	Encoder Count의 방향 반전 여부 0: 정방향 1: 역방향

## 리턴

값	설명
N/A	사용안함

## 예외

예외	설명
N/A	사용안함

## 예제

```

1 set_extenc_mode(1, 2, 20000, 1, 1, 0)
2 # External Encoder channel 1의 동작 모드를 아래와 같이 설정
3 # A상 Count, B상 Direction 사용
4 # Z펄스당 A펄스 카운트는 20000
5 # Z상을 여러 카운트 누적 보상 모드로 사용, S상 사용
6 # Encoder Count의 방향은 정방향으로 설정

```

## 관련 명령어

set\_extenc\_polarity

- [set\\_extenc\\_polarity\(channel, polarity\\_A, polarity\\_B, polarity\\_Z, polarity\\_S\)\(p. 481\)](#)

### 12.1.3 get\_extenc\_count(channel)

#### 기능

해당 엔코더 채널의 현재 카운트 값을 구한다

#### 인수

인수명	자료형	기본값	설명
channel	int	1	엔코더 채널(1, 2) 1: 채널 1 2: 채널 2

#### 리턴

값	설명
count	해당 채널의 Encoder Count 현재 값

#### 예외

예외	설명
N/A	사용안함

#### 예제

```

1   enc_cnt = get_extenc_count(1)
2   # External Encoder channel 1의 현재 count 값을 구함

```

## 관련 명령어

- [set\\_extenc\\_polarity\(channel, polarity\\_A, polarity\\_B, polarity\\_Z, polarity\\_S\)\(p. 481\)](#)
- [set\\_extenc\\_mode\(channel, mode\\_AB, pulse\\_AZ, mode\\_Z, mode\\_S, inverse\\_cnt\)\(p. 482\)](#)

- [clear\\_extenc\\_count\(channel\)\(p. 485\)](#)

### 12.1.4 clear\_extenc\_count(channel)

#### 기능

해당 엔코더 채널의 카운트값을 0으로 초기화한다.

#### 인수

인수명	자료형	기본값	설명
channel	int	1	엔코더 채널(1, 2) 1: 채널 1 2: 채널 2

#### 리턴

값	설명
N/A	사용안함

#### 예외

예외	설명
N/A	사용안함

#### 예제

1	<code>clear_extenc_count(1)</code>
2	<code># External Encoder channel 1의 count 값을 0으로 초기화함</code>

#### 관련 명령어

- [get\\_extenc\\_count\(channel\)\(p. 484\)](#)

## 12.2 Conveyor Tracking

### 12.2.1 set\_conveyor(name)

#### 기능

UI에서 Conveyor 정보를 설정한 경우, 프로그램에서 Conveyor Tracking Application을 시작할 수 있도록 해당 Conveyor 이름으로 ID를 획득하고 작업물 모니터링 명령을 내립니다. 작업물 모니터링은 Conveyor에서 Triggering 되는 작업물에 대해 수행되며 프로그램이 종료될 때까지 계속됩니다.

#### 인수

인수명	자료형	기본값	설명
name	string	-	Conveyor 이름

#### 리턴

값	설명
Conveyor ID	Conveyor 설정 성공 시 Conveyor ID를 리턴
None	Conveyor 설정 실패

#### 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```
1 CONV1 = set_conveyor("conveyor1")
```

## 관련 명령어

- `get_conveyor_obj(conv_id, timeout=None, container_type=DR_FIFO, obj_offset_coord=None)`(p. 491)
- `tracking_conveyor(conv_id, time=0.3)`(p. 495)
- `untracking_conveyor(conv_id, time=0.3)`(p. 497)

### 12.2.2 `set_conveyor_ex(name="", conv_type=0, encoder_channel=1, triggering_mute_time=0.0, count_per_dist=5000, conv_coord=posx(0,0,0,0,0,0), ref=DR_BASE, conv_speed=100.0, speed_filter_size=500, min_dist=0.0, max_dist=1000.0 ...)`

## 기능

Conveyor Tracking Application을 시작할 수 있도록 Conveyor를 설정하고 ID를 획득합니다. 명령이 수행된 이후에는 프로그램이 종료될 때까지 설정된 Conveyor에서 Triggering되는 물체들을 모니터링합니다. Conveyor 정보 setting을 UI를 통해서 설정할 수 없어서 파라미터를 직접 설정해야 할 때 사용합니다.

<sup>1)</sup>M2.4.0 이전 버전 대비 전체 인자의 기본값 추가

<sup>2)</sup>M2.4.0 이전 버전 대비 ref 인자 추가 (world 좌표계 반영)

<sup>3)</sup>M2.4.0 이전 버전 대비 obj\_offset\_coord 인자 삭제, obj\_offset\_coord 인자는 11.2.3 `get_conveyor_obj()` 함수에서만 입력하는 것으로 변경

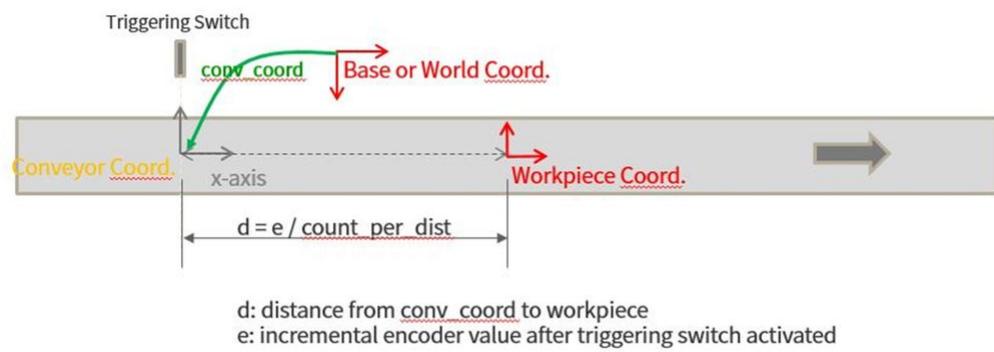
## 인수

인수명	자료형	기본값	설명
name	string	""	Conveyor 이름
conv_type	int	0	Conveyor 타입(0: Linear, 1: Circular)
encoder_channel	int	1	외부 encoder 채널(1, 2)
triggering_mute_time	float	0.0	Triggering 직후에 triggering 신호가 들어와도 triggering(encoder reset, 작업률 초기 시작)을 수행하지 않는 시간(s).

인수명	자료형	기본값	설명
count_per_dist	int	5000	길이 당 엔코더 카운트 환산값 (Linear: count/m, Circular: count/rad)
conv_coord	posx	posx(0,0,0,0,0,0)	컨베이어 고정 좌표계 (Base/World 좌표계 기준, mm, °)
	list (float[6])		
ref	int	DR_BASE	컨베이어 고정 좌표계의 기준 좌표계 (DR_BASE: Base, DR_WORLD: World)
conv_speed	float	100.0	컨베이어 nominal 속도 (Linear: mm/s, Circular: °/s)
speed_filter_size	int	500	컨베이어 속도 Filtering 시 Moving Average Filter Size
min_dist	float	0.0	컨베이어 작업 최소 길이 (Triggering Switch 기준, Linear: mm, Circular: °)
max_dist	float	1000.0	컨베이어 작업 최대 길이 (Triggering Switch 기준, Linear: mm, Circular: °)
watch_window	float	100.0	컨베이어 작업 대기 모니터링 길이 (작업 최소 길이 기준, Linear: mm, Circular: °)
out_tracking_dist	float	10.0	컨베이어 Tracking 해제 여유 구간 길이 (작업 최대 길이 기준, Linear: mm, Circular: °)

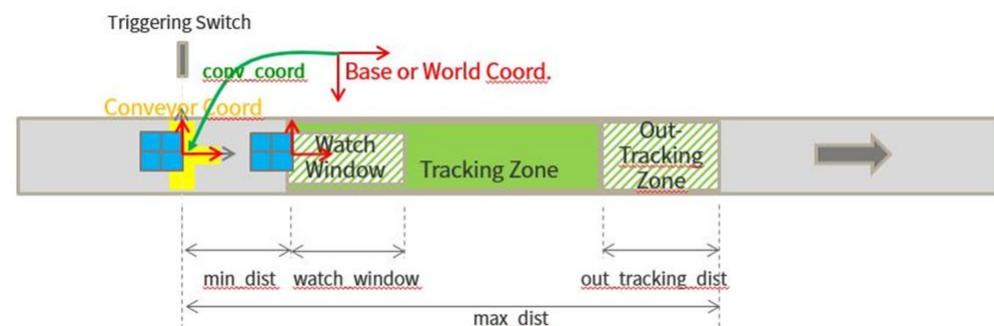
### ❶ 알아두기

1. conv\_type 인자에서 Circular Conveyor는 현재 지원하지 않습니다!
2. set\_conveyor() 또는 set\_conveyor\_ex() 함수 실행 이후 프로그램 종료 전까지, Triggering Switch 를 지나는 모든 작업물은 max\_dist를 지날 때까지 모니터링 됩니다.
3. 단, triggering\_mute\_time을 설정할 경우, 이전 작업물 감지 이후에 해당 시간 동안은 Triggering Switch가 작동되어도 모니터링 목록에 포함시키지 않습니다. Triggering Switch의 노이즈가 있거나 의도적으로 일정 시간 동안 작업물을 배제하고 싶을 때 사용합니다.
4. conv\_coord는 컨베이어에 고정된 좌표계로서 베이스 또는 월드 좌표계 기준의 컨베이어 좌표계입니다. 여기서, **conv\_coord**의 x축은 컨베이어가 흘러가는 방향을 나타냅니다. 컨베이어 작업물이 Triggering Switch를 작동시키는 순간부터 증가된 엔코더 값은 count\_per\_dist 인자를 이용하여 작업물이 이동한 길이로 환산할 수 있고, 이 길이를 conv\_coord의 x축 방향으로 연장하면 기준 좌표계 대비 작업물의 위치를 추적할 수 있습니다.



## [Conveyor 좌표계와 작업물 좌표계]

5. conv\_speed는 컨베이어의 이동 속도입니다. 엔코더로 센싱되는 컨베이어 속도가 이 속도의 200%를 넘을 경우 Info를 주는 용도로만 사용되므로 TP UI를 통해서 측정할 수 없는 경우 대략적인 값을 입력합니다.
6. speed\_filter\_size는 엔코더로부터 컨베이어 속도를 추정할 때 사용되는 moving-average filter의 사이즈입니다. 크기가 클수록 노이즈를 상쇄시킬 수 있지만, 가감속 시 tracking accuracy가 저하될 수 있습니다.
7. 컨베이어 위에 구간은 Watch Window, Tracking Zone, Out-Tracking Zone으로 구분됩니다.
8. Watch Window는 Tracking을 위해서 작업물 좌표계 획득을 하려고 할 때, 해당 구간 안에 있는 작업물만 작업 가능하다고 판단하는 구간입니다. get\_conveyor\_obj() 함수를 호출했을 때, 이 구간 안에 있지 않으면 함수 리턴을 해주지 않고, 작업물이 있으면 get\_conveyor\_obj() 함수 옵션(FIFO, LIFO)에 따라서 작업물 좌표계를 리턴해줍니다.
9. Tracking Zone은 Conveyor Tracking을 수행하는 구간입니다.
10. Out-Tracking Zone은 Tracking을 계속하면 로봇이 고유의 작업 영역 또는 사용자가 설정한 작업 영역을 벗어난다고 판단하여 로봇의 Tracking을 자동으로 종료하게 되는 구간입니다.
11. 3개의 구간은 아래 그림과 같이 4개의 길이(min\_dist, max\_dist, watch\_window, out\_tracking\_dist)로 정의됩니다.



## [Conveyor 구간 및 길이 설명]

## 리턴

값	설명
Conveyor ID	Conveyor 설정 성공 시 Conveyor ID를 리턴
None	Conveyor 설정 실패

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 CONV1 = set_conveyor_ex(name='conveyor_1',
2   conv_type=0, # linear
3   encoder_channel=1, triggering_mute_time=0.0,
4   count_per_dist=5000, # 5000 count/mm)
5   conv_coord=posx(500, 100, 500, 0, -90, 0), ref=DR_BASE,
6   conv_speed=100.0, # conveyor speed: 100 mm/s,
7   speed_filter_size=500, # moving avg. filter size: 500 ms
8   min_dist=100, max_dist=1000, watch_window=200, out_tracking_dist=10)

```

## 관련 명령어

- `get_conveyor_obj(conv_id, timeout=None, container_type=DR_FIFO, obj_offset_coord=None)`(p. 491)
- `tracking_conveyor(conv_id, time=0.3)`(p. 495)
- `untracking_conveyor(conv_id, time=0.3)`(p. 497)

### 12.2.3 get\_conveyor\_obj(conv\_id, timeout=None, container\_type=DR\_FIFO, obj\_offset\_coord=None)

#### 기능

해당 Conveyor로부터 작업 가능한 작업물 좌표계 ID를 리턴합니다. 함수가 불렸을 때, Watch Window에 있는 작업 물 중 설정한 container 규칙에 따라 작업물을 한 개씩 반환해줍니다.

#### 인수

인수명	자료형	기본값	설명
conv_id	int	-	Conveyor ID
timeout	float	None	설정한 시간동안 리턴할 작업물이 없으면 대기 종료하고 함수 리턴
container_type	int	DR_FIFO	작업물 container type(DR_FIFO: first-in/first-out, DR_LIFO: last-in/last-out)
obj_offset_coord	posx	None	컨베이어 고정 좌표계 대비 작업물 좌표계(mm, °)
	list(float[6])		

#### ❶ 알아두기

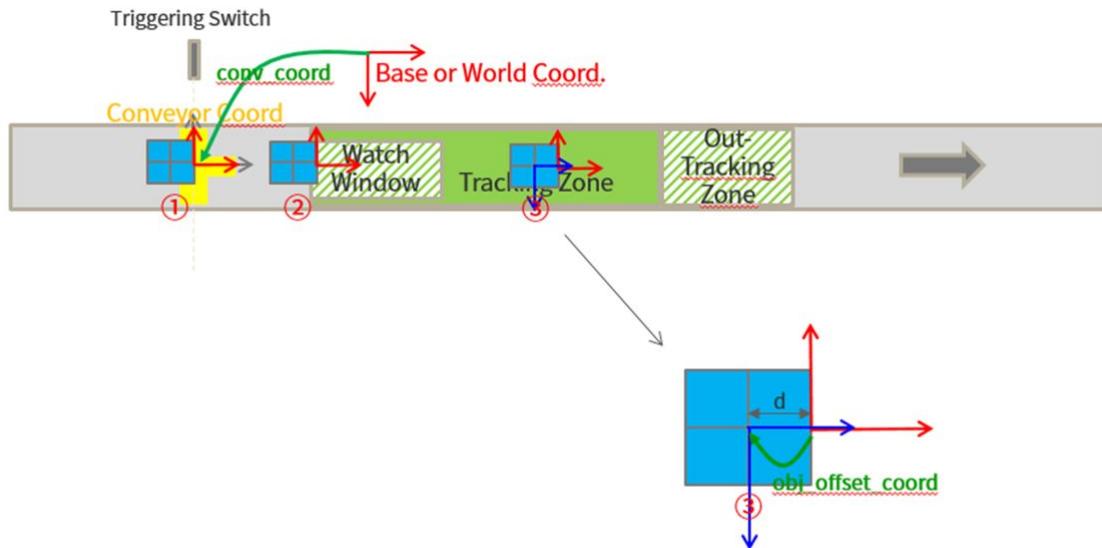
- 해당 함수를 호출했을 때, Watch Window에 있는 작업물을 설정한 Container 규칙에 따라 작업물 좌표계 id를 한 개씩 반환해줍니다. 예를 들어, 아래와 같이 작업물이 배치되어 있는 순간 get\_conveyor\_obj 함수를 호출하면 Watch Window 안에 있는 ②, ③ 작업물이 후보가 됩니다. 이 때, container\_type을 DR\_FIFO로 설정한 경우 Watch Window에 먼저 들어온 ③ 작업물 좌표계 id를 반환하고, DR\_LIFO로 설정한 경우 Watch Window에 나중에 들어온 ② 작업물 좌표계 id를 반환합니다. 만약에 함수 호출 당시에 Watch Window에 작업물이 없으면 최대 timeout 변수에 설정된 시간까지 대기하고 있다가 그 사이에 작업물이 들어오면 해당 id를 반환해줍니다.



[작업물 좌표계 id 반환 규칙 설명]

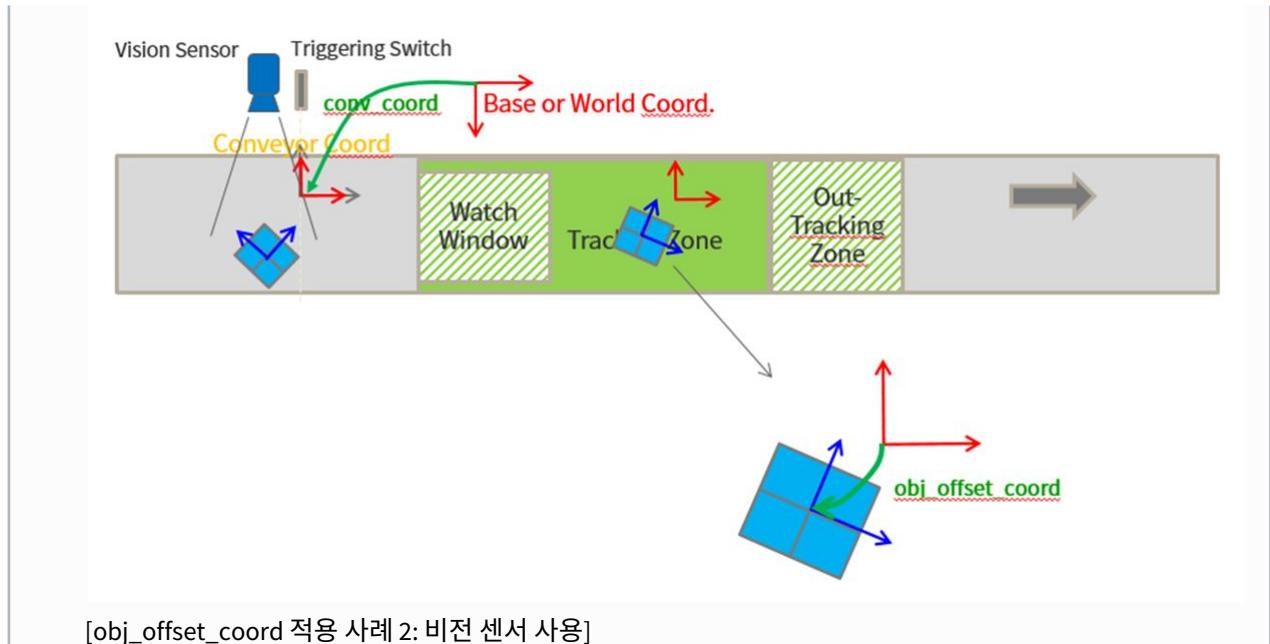
2. `obj_offset_coord`는 작업물 좌표계에 offset을 적용하고 싶을 때 사용합니다. 보통 티칭점을 손쉽게 입력하기 위해서 사용하거나 외부센서(ex. Vision 센서)와 연동해서 작업물의 자세와 위치를 동적으로 바꿔주고 싶을 때 사용합니다. 입력하지 않으면 offset은 0이 적용됩니다.

아래 그림 같은 경우, 작업물 좌표계가 작업물 우측면에 생성되고, Base or World Coord.과 작업물 좌표계가 90도 틀어져있습니다. 이 때, 티칭점의 위치는 작업물 가운데 기준으로, 티칭점의 자세는 Base or World Coord. 기준으로 부여하고 싶은 경우 `obj_offset_coord = posx(-d, 0, 0, -90, 0, 0)`으로 적용하면 됩니다. 이 TP UI를 통해서 티칭점을 획득할 경우 필수적인 부분은 아니지만, drl만을 사용하거나 티칭점을 직접 입력해야 할 경우 필요합니다.



[`obj_offset_coord` 적용 사례 1]

다음으로, 아래 그림 같이 작업물이 컨베이어 진행 방향과 무관한 방향으로 위치가 변하는 경우 또는 작업물의 자세가 바뀌는 경우 엔코더 신호만으로는 작업물의 위치/자세를 알아낼 수 없으므로 비전 센서를 써서 작업물의 추가적인 위치/자세 변화를 감지해야 합니다. 그리고, 여기서 감지한 위치/자세 변화를 동적으로 `obj_offset_coord`에 입력해주면 작업물 좌표계를 이에 맞게 생성합니다.



## 리턴

값	설명
int	CONV_COORD_Conveyor 유저 좌표계 ID (121~150)
음의 정수	timeout 시간이 지나도 반환할 작업물이 없을 경우

### ● 알아두기

반환할 작업물이 없을 경우, timeout 시간이 지날 때까지 함수를 리턴해주지 않습니다. timeout 시간이 지났지만 해당 작업물이 없으면 -1을 반환합니다. 단, timeout 시간을 입력하지 않은 경우는 계속해서 함수를 리턴해주지 않습니다.

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시

예외	설명
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1  ## One object in a cycle
2  CONV1 = set_conveyor('conveyor1')
3
4  movel(posx(100, 100, 50, 0, 0, 0), ref=DR_BASE) # waiting position
5  while True:
6      CONV_COORD_1 = get_conveyor_obj(CONV1)
7      tracking_conveyor(CONV1)
8
9      # synched motion
10     movel(posx(0,0, 50, 0, 0, 0), ref=CONV_COORD_1)
11     movel(posx(0,0, 0, 0, 0, 0), ref=CONV_COORD_1)
12     set_digital_output(DO_GRIPPER, 1)
13     movel(posx(0,0, 50, 0, 0, 0), ref=CONV_COORD_1)
14
15     untracking_conveyor(CONV1)
16
17     movel(posx(100, 100, 50, 0, 0, 0), ref=DR_BASE) # waiting position
18
19     ## Multi objects in a cycle
20     CONV1 = set_conveyor('conveyor1')
21
22     while True:
23         CONV_COORD_1 = get_conveyor_obj(CONV1)
24         tracking_conveyor(CONV1)
25
26         # fist object
27         movel(posx(0,0, 50, 0, 0, 0), ref=CONV_COORD_1)
28         movel(posx(0,0, 0, 0, 0, 0), ref=CONV_COORD_1)
29         set_digital_output(DO_GRIPPER, 1)
30         movel(posx(0,0, 50, 0, 0, 0), ref=CONV_COORD_1)
31
32         # second object
33         CONV_COORD_2 = get_conveyor_obj(CONV1, time_out=10)
34         if CONV_COORD_2 > 0: # -1 if no objects available during time_out
35             movel(posx(0,0, 50, 0, 0, 0), ref=CONV_COORD_2)
36             movel(posx(0,0, 0, 0, 0, 0), ref=CONV_COORD_2)
37             set_digital_output(DO_GRIPPER, 1)
38             movel(posx(0,0, 50, 0, 0, 0), ref=CONV_COORD_2)
39

```

```

40 # first object if you need
41 movel(posx(0,0, 50, 0, 0, 0), ref=CONV_COORD_1)
42 untracking_conveyor(CONV1)
43
44
45 movel(posx(100, 100, 50, 0, 0, 0), ref=DR_BASE)

```

## 관련 명령어

- tracking\_conveyor(conv\_id, time=0.3)(p. 495)
- untracking\_conveyor(conv\_id, time=0.3)(p. 497)

### 12.2.4 tracking\_conveyor(conv\_id, time=0.3)

#### 기능

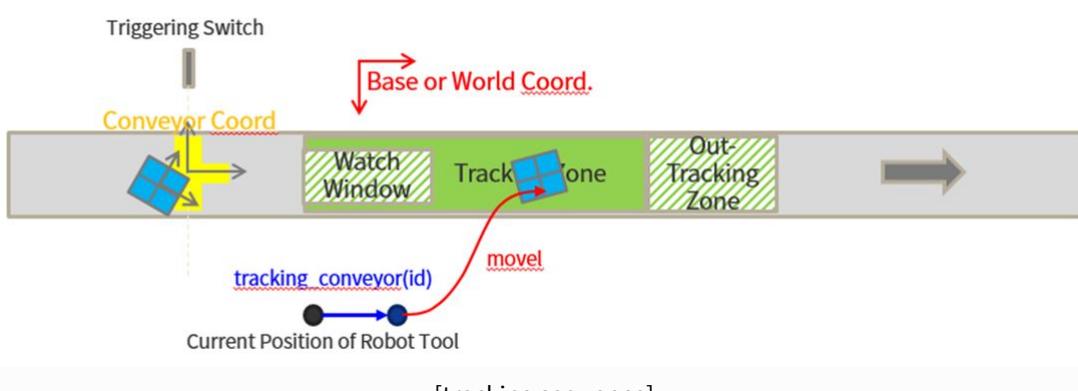
로봇이 가속하여 Conveyor Tracking을 시작합니다.

#### 인수

인수명	자료형	기본값	설명
conv_id	int	-	Conveyor ID
time	float	0.3	Tracking 시작을 위한 가속 시간(sec)

#### 알아두기

tracking\_conveyor 명령이 떨어지면 로봇 현재 위치에서 Conveyor를 tracking하기 시작합니다. 택타임 단축을 위해서 가속 중에도 task motion 호출이 가능하지만, 가속 시간 동안은 오차가 발생할 수 있습니다.



## 리턴

값	설명
0	Conveyor Tracking 성공
음의 정수	가속 과정 중 로봇 작업영역을 벗어날 것으로 예상될 때

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 CONV1 = set_conveyor('conveyor1')
2
3 while True:
4     CONV_COORD_1 = get_conveyor_obj(CONV1)
5
6     tracking_conveyor(CONV1) # start moving to track conveyor
7
8     # task on conveyor
9     movel(posx(0,0, 50, 0, 0, 0), ref=CONV_COORD_1)
10    movel(posx(0,0, 0, 0, 0, 0), ref=CONV_COORD_1)
11    set_digital_output(DO_GRIPPER, 1)
12    movel(posx(0,0, 50, 0, 0, 0), ref=CONV_COORD_1)
13
14    untracking_conveyor(CONV1)
15    obj_count = obj_count + 1

```

## 관련 명령어

- untracking\_conveyor(conv\_id, time=0.3)(p. 497)

### 12.2.5 untracking\_conveyor(conv\_id, time=0.3)

#### 기능

로봇이 감속하여 속도를 0으로 만들고 Conveyor Tracking을 종료합니다.

#### 인수

인수명	자료형	기본값	설명
conv_id	int	-	Conveyor ID
time	float	0.3	Tracking 종료를 위한 감속 시간(sec)

#### ● 알아두기

1. Time 인수를 로봇이 낼 수 있는 최대 감속 시간보다 짧게 입력한 경우에는 시간이 무시되고 최대 감속 시간으로 정지합니다.
2. 택타임 단축을 위해 untracking\_conveyor 함수 뒤에 task motion이 호출되면 감속하면서 task motion을 동시에 수행합니다. (단, 감속 시간 동안 joint motion은 수행할 수 없습니다.)

#### 리턴

값	설명
0	Conveyor Tracking 종료 성공
음의 정수	Conveyor Tracking 종료 모션 중 로봇 작업영역을 벗어날 것으로 예상될 때

#### 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시

예외	설명
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 CONV1 = set_conveyor('conveyor1')
2
3 while True:
4     CONV_COORD_1 = get_conveyor_obj(CONV1)
5     tracking_conveyor(CONV1)
6
7     # task on conveyor
8     movel(posx(0,0, 50, 0, 0, 0), ref=CONV_COORD_1)
9     movel(posx(0,0, 0, 0, 0, 0), ref=CONV_COORD_1)
10    set_digital_output(DO_GRIPPER, 1)
11    movel(posx(0,0, 50, 0, 0, 0), ref=CONV_COORD_1)
12
13    untracking_conveyor(CONV1, 0.1)

```

## 관련 명령어

- [tracking\\_conveyor\(conv\\_id, time=0.3\)\(p. 495\)](#)

## 12.3 Welding

### 12.3.1 app\_weld\_enable\_digital()

#### 기능

통신 인터페이스 방식 용접기능을 활성화합니다. EtherNet/IP 인터페이스만 지원합니다.

#### 리턴

값	설명
0	용접기능 활성화 성공

값	설명
음의값	용접기능 활성화 실패

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

1	app_weld_enable_digital()
2	app_weld_disable_digital()

## 관련 명령어

- [app\\_weld\\_set\\_interface\\_eip\\_r2m\\_process\(welding\\_start=\[0,0,0,0,0,0,0\], robot\\_ready=\[0,0,0,0,0,0,0\], error\\_reset=\[0,0,0,0,0,0,0\]\)\(p. 502\)](#)
- [app\\_weld\\_set\\_interface\\_eip\\_r2m\\_mode\(welding\\_mode=\[0,0,0,0,0,0,0\], s\\_2t=\[0,0,0,0,0,0,0\], pulse\\_mode=\[0,0,0,0,0,0,0\], wm\\_opt1=\[0,0,0,0,0,0,0\]\)\(p. 507\)](#)
- [app\\_weld\\_set\\_interface\\_eip\\_r2m\\_test\(gas\\_test=\[0,0,0,0,0,0,0\], inching\\_plus=\[0,0,0,0,0,0,0\], inching\\_minus=\[0,0,0,0,0,0,0\], blow\\_out\\_torch=\[0,0,0,0,0,0,0\], simulation=\[0,0,0,0,0,0,0\], ts\\_opt1=\[0,0,0,0,0,0,0\], ...\)\(p. 511\)](#)
- [app\\_weld\\_set\\_interface\\_eip\\_r2m\\_condition\(job\\_num=\[0,0,0,0,0,0,0\], synergic\\_id=\[0,0,0,0,0,0,0\], r\\_wire\\_feed\\_speed=\[0,0,0,0,0,0,0\], voltage\\_corret=\[0,0,0,0,0,0,0\], dynamic\\_correct=\[0,0,0,0,0,0,0\]\)\(p. 514\)](#)
- [app\\_weld\\_set\\_interface\\_eip\\_r2m\\_option\(opt1=\[0,0,0,0,0,0,0\], opt2=\[0,0,0,0,0,0,0\], opt3=\[0,0,0,0,0,0,0\], opt4=\[0,0,0,0,0,0,0\], opt5=\[0,0,0,0,0,0,0\], opt6=\[0,0,0,0,0,0,0\], ...\)\(p. 518\)](#)

- `app_weld_set_interface_eip_m2r_process(current_flow=[0,0,0,0,0,0,0,0], process_active=[0,0,0,0,0,0,0,0], main_current=[0,0,0,0,0,0,0,0], machine_ready=[0,0,0,0,0,0,0,0], comm_ready=[0,0,0,0,0,0,0,0])(p. 522)`
- `app_weld_set_interface_eip_m2r_monitoring(welding_voltage=[0,0,0,0,0,0,0,0], welding_current=[0,0,0,0,0,0,0,0], wire_feed_speed=[0,0,0,0,0,0,0,0], wire_stick=[0,0,0,0,0,0,0,0], error=[0,0,0,0,0,0,0,0] ...)(p. 525)`
- `app_weld_set_interface_eip_m2r_other(opt1=[0,0,0,0,0,0,0,0], opt2=[0,0,0,0,0,0,0,0], opt3=[0,0,0,0,0,0,0,0], opt4=[0,0,0,0,0,0,0,0], opt5=[0,0,0,0,0,0,0,0], opt6=[0,0,0,0,0,0,0,0], opt7=[0,0,0,0,0,0,0,0])(p. 529)`
- `app_weld_enable_digital()29`
- `app_weld_set_weld_cond_digital(flag_dry_run=0, vel_target=0, vel_min=0, vel_max=0, welding_mode=0, s_2t=0, pulse_mode=0, wm_opt1=0, simulation=0, ts_opt1=0, ts_opt2=0,...)(p. 533)`
- `app_weld_adj_welding_cond_digital(flag_reset=None, f_target=None, vel_target=None, wv_offset=None, wv_width_ratio=None, dynamic_cor=None, voltage_cor=None, job_number=None, synergic_id=None)(p. 537)`
- `app_weld_disable_digital()(p. 500)`
- `app_weld_weave_cond_trapezoidal(wv_offset=[0,0], wv_ang=0, wv_param=[0,1.5,0,-1.5,0.3,0.1,0.3,0.3,0.1,0.3])(p. 560)`
- `app_weld_weave_cond_zigzag(wv_offset=[0,0], wv_ang=0, wv_param=[3,0.6])(p. 562)`
- `app_weld_weave_cond_circular(wv_offset=[0,0], wv_ang=0, wv_param=[3,3,0.3,0.3])(p. 564)`
- `app_weld_weave_cond_sinusoidal(wv_offset=[0,0], wv_ang=0, wv_param=[3,0.6])(p. 566)`

### 12.3.2 app\_weld\_disable\_digital()

#### 기능

통신 인터페이스 방식 용접기능을 비활성화합니다.

#### 리턴

값	설명
0	성공
음수	실패

<sup>29</sup> [https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/5015159/app\\_weld\\_enable\\_digital](https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/5015159/app_weld_enable_digital)

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 app_weld_enable_digital()
2 app_weld_disable_digital()

```

## 관련 명령어

- [app\\_weld\\_set\\_interface\\_eip\\_r2m\\_process\(welding\\_start=\[0,0,0,0,0,0,0,0\], robot\\_ready=\[0,0,0,0,0,0,0,0\], error\\_reset=\[0,0,0,0,0,0,0,0\]\)\(p. 502\)](#)
- [app\\_weld\\_set\\_interface\\_eip\\_r2m\\_mode\(welding\\_mode=\[0,0,0,0,0,0,0,0\], s\\_2t=\[0,0,0,0,0,0,0,0\], pulse\\_mode=\[0,0,0,0,0,0,0,0\], wm\\_opt1=\[0,0,0,0,0,0,0,0\]\)\(p. 507\)](#)
- [app\\_weld\\_set\\_interface\\_eip\\_r2m\\_test\(gas\\_test=\[0,0,0,0,0,0,0,0\], inching\\_plus=\[0,0,0,0,0,0,0,0\], inching\\_minus=\[0,0,0,0,0,0,0,0\], blow\\_out\\_torch=\[0,0,0,0,0,0,0,0\], simulation=\[0,0,0,0,0,0,0,0\], ts\\_opt1=\[0,0,0,0,0,0,0,0\], ...\)\(p. 511\)](#)
- [app\\_weld\\_set\\_interface\\_eip\\_r2m\\_condition\(job\\_num=\[0,0,0,0,0,0,0,0\], synergic\\_id=\[0,0,0,0,0,0,0,0\], r\\_wire\\_feed\\_speed=\[0,0,0,0,0,0,0,0\], voltage\\_corret=\[0,0,0,0,0,0,0,0\], dynamic\\_correct=\[0,0,0,0,0,0,0,0\]\)\(p. 514\)](#)
- [app\\_weld\\_set\\_interface\\_eip\\_r2m\\_option\(opt1=\[0,0,0,0,0,0,0,0\], opt2=\[0,0,0,0,0,0,0,0\], opt3=\[0,0,0,0,0,0,0,0\], opt4=\[0,0,0,0,0,0,0,0\], opt5=\[0,0,0,0,0,0,0,0\], opt6=\[0,0,0,0,0,0,0,0\], ...\)\(p. 518\)](#)
- [app\\_weld\\_set\\_interface\\_eip\\_m2r\\_process\(current\\_flow=\[0,0,0,0,0,0,0,0\], process\\_active=\[0,0,0,0,0,0,0,0\], main\\_current=\[0,0,0,0,0,0,0,0\], machine\\_ready=\[0,0,0,0,0,0,0,0\], comm\\_ready=\[0,0,0,0,0,0,0,0\]\)\(p. 522\)](#)
- [app\\_weld\\_set\\_interface\\_eip\\_m2r\\_monitoring\(welding\\_voltage=\[0,0,0,0,0,0,0,0\], welding\\_current=\[0,0,0,0,0,0,0,0\], wire\\_feed\\_speed=\[0,0,0,0,0,0,0,0\], wire\\_stick=\[0,0,0,0,0,0,0,0\], error=\[0,0,0,0,0,0,0,0\] ...\)\(p. 525\)](#)

- `app_weld_set_interface_eip_m2r_other(opt1=[0,0,0,0,0,0,0], opt2=[0,0,0,0,0,0,0], opt3=[0,0,0,0,0,0,0], opt4=[0,0,0,0,0,0,0], opt5=[0,0,0,0,0,0,0], opt6=[0,0,0,0,0,0,0], opt7=[0,0,0,0,0,0,0],...)(p. 529)`
- `app_weld_disable_digital()30`
- `app_weld_set_weld_cond_digital(flag_dry_run=0, vel_target=0, vel_min=0, vel_max=0, welding_mode=0, s_2t=0, pulse_mode=0, wm_opt1=0, simulation=0, ts_opt1=0, ts_opt2=0,...)(p. 533)`
- `app_weld_adj_welding_cond_digital(flag_reset=None, f_target=None, vel_target=None, wv_offset=None, wv_width_ratio=None, dynamic_cor=None, voltage_cor=None, job_number=None, synergic_id=None)(p. 537)`
- `app_weld_disable_digital()31`
- `app_weld_weave_cond_trapezoidal(wv_offset=[0,0], wv_ang=0, wv_param=[0,1.5,0,-1.5,0.3,0.1,0.3,0.3,0.1,0.3])(p. 560)`
- `app_weld_weave_cond_zigzag(wv_offset=[0,0], wv_ang=0, wv_param=[3,0.6])(p. 562)`
- `app_weld_weave_cond_circular(wv_offset=[0,0], wv_ang=0, wv_param=[3,3,0.3,0.3])(p. 564)`
- `app_weld_weave_cond_sinusoidal(wv_offset=[0,0], wv_ang=0, wv_param=[3,0.6])(p. 566)`

### 12.3.3 `app_weld_set_interface_eip_r2m_process(welding_start=[0,0,0,0,0,0,0,0], robot_ready=[0,0,0,0,0,0,0,0], error_reset=[0,0,0,0,0,0,0,0])`

#### 기능

EtherNet/IP 통신을 지원하는 용접기를 사용하기 위한 통신 인터페이스 설정을 합니다. 로봇제어기에서 용접기로 보내는 통신 데이터 중, 용접 수행을 위해 로봇제어기와 용접기 간 연동 신호에 대한 인터페이스를 설정합니다. 아래 설정값과 관련된 세부 사항은 해당 용접기의 통신 시그널 데이터시트를 참고하여 기입하기 바랍니다.

#### 알아두기

EtherNet/IP 통신방식 용접기를 이용한 정상적인 용접기능 수행을 위해서는 8종의 인터페이스 설정 명령어의 설정이 모두 완료되어야 합니다. `app_weld_set_interface_eip_r2m_process()`, `app_weld_set_interface_eip_r2m_mode()`, `app_weld_set_interface_eip_r2m_test()`, `app_weld_set_interface_eip_r2m_condition()`, `app_weld_set_interface_eip_r2m_option()`, `app_weld_set_interface_eip_m2r_process()`, `app_weld_set_interface_eip_m2r_monitoring()`, `app_weld_set_interface_eip_m2r_other()`

<sup>30</sup> [https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/5015163/app\\_weld\\_disable\\_digital](https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/5015163/app_weld_disable_digital)

<sup>31</sup> [https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/5015163/app\\_weld\\_disable\\_digital](https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/5015163/app_weld_disable_digital)

## 인수

인수명	자료형	기본값	설명
welding_start	아래참조	아래참조	용접시작 명령 (용접기별 사양)
robot_ready			로봇 상태 (용접기별 사양)
error_reset			용접기 발생 에러 초기화 (용접기별 사양)

인수 자료형, 기본값, 설명은 아래와 동일

인수명	자료형	기본값	설명
	list(int[7])	0	미사용: 0 사용: 1
		0	데이터 타입(on/off: 0, 선택: 1, 값: 2)
		0	데이터 자릿수(1: 0, 0.1: 1, 0.01: 2)
		0	통신 데이터 위치 (byte): 1~255
		0	통신 데이터 위치 (bit): 1~255
		0	데이터 사이즈 1-bit(disable Low): 0 1-bit(disable High): 1 2-bit: 2 4-bit: 3 8-bit(byte): 4 15-bit: 5 16-bit(short): 6 32-bit(int): 7

인수명	자료형	기본값	설명
	list(float[2])	0	유효한 데이터 사이즈 값(bit)
		0	데이터 최소값
		0	데이터 최대값

### 알아두기

통신 인터페이스 설정 예시(EWM 용접기)

데이터 타입(on/off: 0)인 경우 : On/Off 중 선택하는 항목

Ewm 용접기 데이터 시트

Byte no.	Bit no.	Function/description	Bit assignment
0	4	Start signal welding process	0 switched off 1 switched on

사양 정보 기입 방법

항목	설정값
사용 여부	1 (사용)
데이터 타입	0 (on/off)
데이터 자릿수	0 (1)
통신 데이터 위치(byte)	0
통신 데이터 위치(bit)	4
데이터 크기	0 (1-bit, disable Low)
유효한 데이터 사이즈	1 (1 bit)
데이터 최소값	0
데이터 최대값	1

데이터 타입(선택: 1)인 경우 : 정수형 자료 중 하나를 선택하는 경우

Ewm 용접기 데이터 시트

<b>Byte no.</b>	<b>Bit no.</b>	<b>Function/description</b>	<b>Bit assignment</b>
3	0-7	Selection JOB	Range 1-255

사양 정보 기입 방법

<b>항목</b>	<b>설정값</b>
사용 여부	1 (사용)
데이터 타입	1 (선택)
데이터 자릿수	0 (1)
통신 데이터 위치(byte)	3
통신 데이터 위치(bit)	0
데이터 크기	4 (8-bit)
유효한 데이터 사이즈	8 (8 bit)
데이터 최소값	0
데이터 최대값	255

데이터 타입(값: 2)인 경우 : 실수값을 입력하는 경우

Ewm 용접기 데이터 시트

<b>Byte no.</b>	<b>Bit no.</b>	<b>Function/description</b>	<b>Bit assignment</b>
6	0-15	Welding voltage(current actual value)	0 to 0x7FFF (High-Byte first) equivalent to 0.0V to 100.0V

사양 정보 기입 방법

<b>항목</b>	<b>설정값</b>
사용 여부	1 (사용)
데이터 타입	2 (값)
데이터 자릿수	1 (0.1)
통신 데이터 위치(byte)	6
통신 데이터 위치(bit)	0
데이터 크기	6 (16-bit)
유효한 데이터 사이즈	15 (15 bit)

항목	설정값
데이터 최소값	0.0 (V)
데이터 최대값	100.0 (V)

• 데이터 타입이 2(값)인 경우 해당 데이터가 표현 할 수 있는 값과 실제 물리적인 값과의 매칭을 위해 유효한 데이터 사이즈(0x7FFF → 15 bit), 데이터 최소값(0.0V), 데이터 최대 값(100.0V) 정보를 추가적으로 기입하여야 됩니다.

## 리턴

값	설명
0	성공
음수	실패

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

1	app_weld_set_interface_eip_r2m_process(welding_start=[1,0,0,0,4,0,1,0,0], robot_ready=[1,0,0,0,5,0,1,0,0], error_reset=[1,0,0,1,4,0,1,0,0])
---	---

## 관련 명령어

- app\_weld\_set\_interface\_eip\_r2m\_process(welding\_start=[0,0,0,0,0,0,0,0], robot\_ready=[0,0,0,0,0,0,0,0], error\_reset=[0,0,0,0,0,0,0,0])<sup>32</sup>
- app\_weld\_set\_interface\_eip\_r2m\_mode(welding\_mode=[0,0,0,0,0,0,0,0], s\_2t=[0,0,0,0,0,0,0,0], pulse\_mode=[0,0,0,0,0,0,0,0],wm\_opt1=[0,0,0,0,0,0,0,0])(p. 507)
- app\_weld\_set\_interface\_eip\_r2m\_test(gas\_test=[0,0,0,0,0,0,0,0], inching\_plus=[0,0,0,0,0,0,0,0], inching\_minus=[0,0,0,0,0,0,0,0], blow\_out\_torch=[0,0,0,0,0,0,0,0], simulation=[0,0,0,0,0,0,0,0], ts\_opt1=[0,0,0,0,0,0,0,0], ...)(p. 511)
- app\_weld\_set\_interface\_eip\_r2m\_condition(job\_num=[0,0,0,0,0,0,0,0], synergic\_id=[0,0,0,0,0,0,0,0], r\_wire\_feed\_speed=[0,0,0,0,0,0,0,0], voltage\_corret=[0,0,0,0,0,0,0,0], dynamic\_correct=[0,0,0,0,0,0,0,0])(p. 514)
- app\_weld\_set\_interface\_eip\_r2m\_option(opt1=[0,0,0,0,0,0,0,0], opt2=[0,0,0,0,0,0,0,0], opt3=[0,0,0,0,0,0,0,0], opt4=[0,0,0,0,0,0,0,0], opt5=[0,0,0,0,0,0,0,0], opt6=[0,0,0,0,0,0,0,0],...)(p. 518)
- app\_weld\_set\_interface\_eip\_m2r\_process(current\_flow=[0,0,0,0,0,0,0,0], process\_active=[0,0,0,0,0,0,0,0], main\_current=[0,0,0,0,0,0,0,0], machine\_ready=[0,0,0,0,0,0,0,0], comm\_ready=[0,0,0,0,0,0,0,0])(p. 522)
- app\_weld\_set\_interface\_eip\_m2r\_monitoring(welding\_voltage=[0,0,0,0,0,0,0,0], welding\_current=[0,0,0,0,0,0,0,0], wire\_feed\_speed=[0,0,0,0,0,0,0,0], wire\_stick=[0,0,0,0,0,0,0,0], error=[0,0,0,0,0,0,0,0] ...)(p. 525)
- app\_weld\_set\_interface\_eip\_m2r\_other(opt1=[0,0,0,0,0,0,0,0], opt2=[0,0,0,0,0,0,0,0], opt3=[0,0,0,0,0,0,0,0], opt4=[0,0,0,0,0,0,0,0], opt5=[0,0,0,0,0,0,0,0], opt6=[0,0,0,0,0,0,0,0], opt7=[0,0,0,0,0,0,0,0],...)(p. 529)

### 12.3.4 app\_weld\_set\_interface\_eip\_r2m\_mode(welding\_mode=[0,0,0,0,0,0,0,0], s\_2t=[0,0,0,0,0,0,0,0], pulse\_mode=[0,0,0,0,0,0,0,0],wm\_opt1=[0,0,0,0,0,0,0,0])

#### 기능

EtherNet/IP 통신을 지원하는 용접기를 사용하기 위한 통신 인터페이스 설정을 합니다. 로봇제어기에서 용접기로 보내는 통신 데이터 중, 용접모드와 관련한 인터페이스를 설정합니다. 추가적으로 필요한 모드 선택 기능은 옵션 항목 (wm\_opt1)을 통해 추가할 수 있습니다. 아래 설정값과 관련된 세부 사항은 해당 용접기의 통신 시그널 데이터시트를 참고하여 기입하기 바랍니다.

#### 알아두기

EtherNet/IP 통신방식 용접기를 이용한 정상적인 용접기능 수행을 위해서는 8종의 인터페이스 설정 명령어의 설정이 모두 완료되어야 합니다.

<sup>32</sup><https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/5015167/>

app\_weld\_set\_interface\_eip\_r2m\_process+welding\_start+0+0+0+0+0+0+0+0+robot\_ready+0+0+0+0+0+0+0+error\_reset+0+0+0+0+0+0+0+0

```
app_weld_set_interface_eip_r2m_process(), app_weld_set_interface_eip_r2m_mode(),
app_weld_set_interface_eip_r2m_test(), app_weld_set_interface_eip_r2m_condition(),
app_weld_set_interface_eip_r2m_option(), app_weld_set_interface_eip_m2r_process(),
app_weld_set_interface_eip_m2r_monitoring(), app_weld_set_interface_eip_m2r_other()
```

## 인수

인수명	자료형	기본값	설명
welding_mode	아래참조	아래참조	용접모드 (용접기별사양)
s_2t			latched/non-latched모드 (용접기별사양)
pulse_mode			펄스모드 (용접기별 사양)
wm_opt1			옵션모드 (용접기별 사양)

인수 자료형, 기본값, 설명은 아래와 동일

인수명	자료형	기본값	설명
	list(int[7])	0	미사용: 0 사용: 1
		0	데이터 타입(on/off: 0, 선택: 1, 값: 2)
		0	데이터 자릿수(1: 0, 0.1: 1, 0.01: 2)
		0	통신 데이터 위치(byte): 1~255
		0	통신 데이터 위치(bit): 1~255

인수명	자료형	기본값	설명
		0	데이터 사이즈 1-bit(disable Low): 0 1-bit(disable High): 1 2-bit: 2 4-bit: 3 8-bit(byte): 4 15-bit: 5 16-bit(short): 6 32-bit(int): 7
		0	유효한 데이터 사이즈 값(bit)
	list(float[2])	0	데이터 최소값
		0	데이터 최대값

### ❶ 알아두기

데이터 타입(0~2)별 인터페이스 설정 예시는 `app_weld_set_interface_eip_r2m_process()` 절을 참고하세요.

### 리턴

값	설명
0	성공
음수	실패

### 예외

예외	설명
<code>DR_Error</code> ( <code>DR_ERROR_TYPE</code> )	인수들의 데이터형 오류 시

예외	설명
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

예제

```
1 app_weld_set_interface_eip_r2m_mode(welding_mode=[1,1,0,0,0,2,2,0,3],  
s_2t=[0,0,0,0,0,0,0,0,0], pulse_mode=[1,1,0,0,2,0,1,0,1],wm_opt1=[0,0,0,0,0,  
)
```

## 관련 명령어

- app\_weld\_set\_interface\_eip\_r2m\_process(welding\_start=[0,0,0,0,0,0,0,0], robot\_ready=[0,0,0,0,0,0,0,0], error\_reset=[0,0,0,0,0,0,0,0])(p. 502)
  - app\_weld\_set\_interface\_eip\_r2m\_mode(welding\_mode=[0,0,0,0,0,0,0,0], s\_2t=[0,0,0,0,0,0,0,0], pulse\_mode=[0,0,0,0,0,0,0,0],wm\_opt1=[0,0,0,0,0,0,0,0])<sup>33</sup>
  - app\_weld\_set\_interface\_eip\_r2m\_test(gas\_test=[0,0,0,0,0,0,0,0], inching\_plus=[0,0,0,0,0,0,0,0], inching\_minus=[0,0,0,0,0,0,0,0], blow\_out\_torch=[0,0,0,0,0,0,0,0], simulation=[0,0,0,0,0,0,0,0], ts\_opt1=[0,0,0,0,0,0,0,0], ...)(p. 511)
  - app\_weld\_set\_interface\_eip\_r2m\_condition(job\_num=[0,0,0,0,0,0,0,0], synergic\_id=[0,0,0,0,0,0,0,0], r\_wire\_feed\_speed=[0,0,0,0,0,0,0,0], voltage\_corret=[0,0,0,0,0,0,0,0.0], dynamic\_correct=[0,0,0,0,0,0,0,0])(p. 514)
  - app\_weld\_set\_interface\_eip\_r2m\_option(opt1=[0,0,0,0,0,0,0,0], opt2=[0,0,0,0,0,0,0,0], opt3=[0,0,0,0,0,0,0,0], opt4=[0,0,0,0,0,0,0,0], opt5=[0,0,0,0,0,0,0,0], opt6=[0,0,0,0,0,0,0,0], ...)(p. 518)
  - app\_weld\_set\_interface\_eip\_m2r\_process(current\_flow=[0,0,0,0,0,0,0,0], process\_active=[0,0,0,0,0,0,0,0], main\_current=[0,0,0,0,0,0,0,0], machine\_ready=[0,0,0,0,0,0,0,0], comm\_ready=[0,0,0,0,0,0,0,0])(p. 522)
  - app\_weld\_set\_interface\_eip\_m2r\_monitoring(welding\_voltage=[0,0,0,0,0,0,0,0], welding\_current=[0,0,0,0,0,0,0,0], wire\_feed\_speed=[0,0,0,0,0,0,0,0], wire\_stick=[0,0,0,0,0,0,0,0], error=[0,0,0,0,0,0,0,0] ...)(p. 525)

<https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/5015171/>

- `app_weld_set_interface_eip_m2r_other(opt1=[0,0,0,0,0,0,0], opt2=[0,0,0,0,0,0,0], opt3=[0,0,0,0,0,0,0], opt4=[0,0,0,0,0,0,0], opt5=[0,0,0,0,0,0,0], opt6=[0,0,0,0,0,0,0], opt7=[0,0,0,0,0,0,0], ...)(p. 529)`

### 12.3.5 `app_weld_set_interface_eip_r2m_test(gas_test=[0,0,0,0,0,0,0,0], inching_plus=[0,0,0,0,0,0,0,0], inching_minus=[0,0,0,0,0,0,0,0], blow_out_torch=[0,0,0,0,0,0,0,0], simulation=[0,0,0,0,0,0,0,0], ts_opt1=[0,0,0,0,0,0,0,0], ...)`

#### 기능

EtherNet/IP 통신을 지원하는 용접기를 사용하기 위한 통신 인터페이스 설정을 합니다. 로봇제어기에서 용접기로 보내는 통신 데이터 중, 테스트 신호 설정과 관련한 인터페이스를 설정합니다. 추가적인 테스트 신호 관련 설정은 옵션 항목(ts\_opt1, ts\_opt2)을 통해 추가할 수 있습니다. 아래 설정값과 관련된 세부 사항은 해당 용접기의 통신 시그널 데이터시트를 참고하여 기입하기 바랍니다

#### 알아두기

EtherNet/IP 통신방식 용접기를 이용한 정상적인 용접기능 수행을 위해서는 8종의 인터페이스 설정 명령어의 설정이 모두 완료되어야 합니다.

```
app_weld_set_interface_eip_r2m_process(), app_weld_set_interface_eip_r2m_mode(),
app_weld_set_interface_eip_r2m_test(), app_weld_set_interface_eip_r2m_condition(),
app_weld_set_interface_eip_r2m_option(), app_weld_set_interface_eip_m2r_process(),
app_weld_set_interface_eip_m2r_monitoring(), app_weld_set_interface_eip_m2r_other()
```

#### 인수

인수명	자료형	기본값	설명
gas_test	아래참조	아래참조	가스테스트 신호 (용접기별사양)
Inching_plus			정방향인칭 신호 (용접기별사양)
Inching_minus			역방향인칭 신호 (용접기별사양)
blow_out_torch			토치청소 신호 (용접기별사양)
simulation			모의용접 신호 (용접기별사양)
ts_opt1			옵션 신호 (용접기별사양)
ts_opt2			옵션 신호 (용접기별사양)

인수 자료형, 기본값, 설명은 아래와 동일

인수명	자료형	기본값	설명
	list(int[7])	0	미사용: 0 사용: 1
		0	데이터 타입(on/off: 0, 선택: 1, 값: 2)
		0	데이터 자릿수(1: 0, 0.1: 1, 0.01: 2)
		0	통신 데이터 위치 (byte): 1~255
		0	통신 데이터 위치 (bit): 1~255
		0	데이터 사이즈 1-bit(disable Low): 0 1-bit(disable High): 1 2-bit: 2 4-bit: 3 8-bit(byte): 4 15-bit: 5 16-bit(short): 6 32-bit(int): 7
		0	유효한 데이터 사이즈 값(bit)
	list(float[2])	0	데이터 최소값
		0	데이터 최대값

### ❶ 알아두기

데이터 타입(0~2)별 인터페이스 설정 예시는 `app_weld_set_interface_eip_r2m_process()` 절을 참고하세요

## 리턴

값	설명
0	성공
음수	실패

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```
1 app_weld_set_interface_eip_r2m_test(gas_test=[1,0,0,0,6,0,1,0,0],  
inking_plus=[1,0,0,1,0,0,1,0,0], inking_minus=[1,0,0,1,2,0,1,0,0],  
blow_out_torch=[1,0,0,0,7,0,1,0,0], simulation=[0,0,0,1,7,0,1,0,0],  
ts_opt1=[0,0,0,0,0,0,0,0], ts_opt2=[0,0,0,0,0,0,0,0])
```

## 관련 명령어

- `app_weld_set_interface_eip_r2m_process(welding_start=[0,0,0,0,0,0,0,0], robot_ready=[0,0,0,0,0,0,0], error_reset=[0,0,0,0,0,0,0,0])(p. 502)`
- `app_weld_set_interface_eip_r2m_mode(welding_mode=[0,0,0,0,0,0,0,0], s_2t=[0,0,0,0,0,0,0,0], pulse_mode=[0,0,0,0,0,0,0,0],wm_opt1=[0,0,0,0,0,0,0,0])(p. 507)`

- app\_weld\_set\_interface\_eip\_r2m\_test(gas\_test=[0,0,0,0,0,0,0,0], inching\_plus=[0,0,0,0,0,0,0,0], inching\_minus=[0,0,0,0,0,0,0,0], blow\_out\_torch=[0,0,0,0,0,0,0,0], simulation=[0,0,0,0,0,0,0,0], ts\_opt1=[0,0,0,0,0,0,0,0], ...)<sup>34</sup>
  - app\_weld\_set\_interface\_eip\_r2m\_condition(job\_num=[0,0,0,0,0,0,0,0], synergic\_id=[0,0,0,0,0,0,0,0], r\_wire\_feed\_speed=[0,0,0,0,0,0,0,0], voltage\_corret=[0,0,0,0,0,0,0,0.0], dynamic\_correct=[0,0,0,0,0,0,0,0])(p. 514)
  - app\_weld\_set\_interface\_eip\_r2m\_option(opt1=[0,0,0,0,0,0,0,0], opt2=[0,0,0,0,0,0,0,0], opt3=[0,0,0,0,0,0,0,0], opt4=[0,0,0,0,0,0,0,0], opt5=[0,0,0,0,0,0,0,0], opt6=[0,0,0,0,0,0,0,0], ...)(p. 518)
  - app\_weld\_set\_interface\_eip\_m2r\_process(current\_flow=[0,0,0,0,0,0,0,0], process\_active=[0,0,0,0,0,0,0,0], main\_current=[0,0,0,0,0,0,0,0], machine\_ready=[0,0,0,0,0,0,0,0], comm\_ready=[0,0,0,0,0,0,0,0])(p. 522)
  - app\_weld\_set\_interface\_eip\_m2r\_monitoring(welding\_voltage=[0,0,0,0,0,0,0,0], welding\_current=[0,0,0,0,0,0,0,0], wire\_feed\_speed=[0,0,0,0,0,0,0,0], wire\_stick=[0,0,0,0,0,0,0,0], error=[0,0,0,0,0,0,0,0] ...)(p. 525)
  - app\_weld\_set\_interface\_eip\_m2r\_other(opt1=[0,0,0,0,0,0,0,0], opt2=[0,0,0,0,0,0,0,0], opt3=[0,0,0,0,0,0,0,0], opt4=[0,0,0,0,0,0,0,0], opt5=[0,0,0,0,0,0,0,0], opt6=[0,0,0,0,0,0,0,0], opt7=[0,0,0,0,0,0,0,0], ...)(p. 529)

**12.3.6 app\_weld\_set\_interface\_eip\_r2m\_condition(job\_num=[0,0,0,0,0,0,0,0,0], synergic\_id=[0,0,0,0,0,0,0,0,0], r\_wire\_feed\_speed=[0,0,0,0,0,0,0,0,0], voltage\_corret=[0,0,0,0,0,0,0,0,0], dynamic\_correct=[0,0,0,0,0,0,0,0,0])**

기능

EtherNet/IP 통신을 지원하는 용접기를 사용하기 위한 통신 인터페이스 설정을 합니다. 로봇제어기에서 용접기로 보내는 통신 데이터 중, 용접 조건 설정과 관련한 인터페이스를 설정합니다. 아래 설정값과 관련된 세부 사항은 해당 용접기의 통신 시그널 데이터시트를 참고하여 기입하기 바랍니다.

## 알아두기

EtherNet/IP 통신방식 용접기를 이용한 정상적인 용접기능 수행을 위해서는 8종의 인터페이스 설정 명령어의 설정이 모두 완료되어야 합니다. `app_weld_set_interface_eip_r2m_process()`,  
`app_weld_set_interface_eip_r2m_mode()`, `app_weld_set_interface_eip_r2m_test()`,  
`app_weld_set_interface_eip_r2m_condition()`, `app_weld_set_interface_eip_r2m_option()`,  
`app_weld_set_interface_eip_m2r_process()`, `app_weld_set_interface_eip_m2r_monitoring()`,  
`app_weld_set_interface_eip_m2r_other()`

[34https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/5015175/](https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/5015175/)

## 인수

인수명	자료형	기본값	설명
job_num	아래참조	아래참조	JOB번호 (용접기별사양)
synergic_id			시너직(SYNERGIC) 번호 (용접기별사양)
r_wire_feed_speed			와이어피딩속도조절 (용접기별사양)
voltage_correct			전압조정 (용접기별사양)
dynamic_correct			동적변수조정 (용접기별사양)

인수 자료형, 기본값, 설명은 아래와 동일

인수명	자료형	기본값	설명
	list(int[7])	0	미사용: 0 사용: 1
		0	데이터 타입(on/off: 0, 선택: 1, 값: 2)
		0	데이터 자릿수(1: 0, 0.1: 1, 0.01: 2)
		0	통신 데이터 위치(byte): 1~255
		0	통신 데이터 위치(bit): 1~255

인수명	자료형	기본값	설명
		0	데이터 사이즈 1-bit(disable Low): 0 1-bit(disable High): 1 2-bit: 2 4-bit: 3 8-bit(byte): 4 15-bit: 5 16-bit(short): 6 32-bit(int): 7
		0	유효한 데이터 사이즈 값(bit)
	list(float[2])	0	데이터 최소값
		0	데이터 최대값

### ❶ 알아두기

데이터 타입(0~2)별 인터페이스 설정 예시는 `app_weld_set_interface_eip_r2m_process()` 절을 참고하세요

### 리턴

값	설명
0	성공
음수	실패

### 예외

예외	설명
<code>DR_Error</code> ( <code>DR_ERROR_TYPE</code> )	인수들의 데이터형 오류 시

예외	설명
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

1	app_weld_set_interface_eip_r2m_condition(job_num=[1, 1, 0, 3, 0, 4, 8, 0, 255], synergic_id=[1, 1, 0, 2, 0, 3, 4, 0, 15], r_wire_feed_speed=[1, 2, 1, 6, 0, 6, 15, 0.0, 25.0], voltage_corret=[1, 2, 1, 8, 0, 6, 15, -10.0, 10.0], dynamic_correct=[1, 2, 0, 10, 0.5, -40, 40])
---	---

## 관련 명령어

- app\_weld\_set\_interface\_eip\_r2m\_process(welding\_start=[0,0,0,0,0,0,0,0], robot\_ready=[0,0,0,0,0,0,0,0], error\_reset=[0,0,0,0,0,0,0,0])(p. 502)
- app\_weld\_set\_interface\_eip\_r2m\_mode(welding\_mode=[0,0,0,0,0,0,0,0], s\_2t=[0,0,0,0,0,0,0,0], pulse\_mode=[0,0,0,0,0,0,0,0],wm\_opt1=[0,0,0,0,0,0,0,0])(p. 507)
- app\_weld\_set\_interface\_eip\_r2m\_test(gas\_test=[0,0,0,0,0,0,0,0], inching\_plus=[0,0,0,0,0,0,0,0], inching\_minus=[0,0,0,0,0,0,0,0], blow\_out\_torch=[0,0,0,0,0,0,0,0], simulation=[0,0,0,0,0,0,0,0], ts\_opt1=[0,0,0,0,0,0,0,0], ...)(p. 511)
- app\_weld\_set\_interface\_eip\_r2m\_condition(job\_num=[0,0,0,0,0,0,0,0], synergic\_id=[0,0,0,0,0,0,0,0], r\_wire\_feed\_speed=[0,0,0,0,0,0,0,0], voltage\_corret=[0,0,0,0,0,0,0,0], dynamic\_correct=[0,0,0,0,0,0,0,0])<sup>35</sup>
- app\_weld\_set\_interface\_eip\_r2m\_option(opt1=[0,0,0,0,0,0,0,0], opt2=[0,0,0,0,0,0,0,0], opt3=[0,0,0,0,0,0,0,0], opt4=[0,0,0,0,0,0,0,0], opt5=[0,0,0,0,0,0,0,0], opt6=[0,0,0,0,0,0,0,0], ...)(p. 518)
- app\_weld\_set\_interface\_eip\_m2r\_process(current\_flow=[0,0,0,0,0,0,0,0], process\_active=[0,0,0,0,0,0,0,0], main\_current=[0,0,0,0,0,0,0,0], machine\_ready=[0,0,0,0,0,0,0,0], comm\_ready=[0,0,0,0,0,0,0,0])(p. 522)
- app\_weld\_set\_interface\_eip\_m2r\_monitoring(welding\_voltage=[0,0,0,0,0,0,0,0], welding\_current=[0,0,0,0,0,0,0,0], wire\_feed\_speed=[0,0,0,0,0,0,0,0], wire\_stick=[0,0,0,0,0,0,0,0], error=[0,0,0,0,0,0,0,0] ...)(p. 525)

<sup>35</sup><https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/5015179/>

app\_weld\_set\_interface\_eip\_r2m\_condition+job\_num+0+0+0+0+0+0+0+0+synergic\_id+0+0+0+0+0+0+0+r\_wire\_feed\_s  
peed+0+0+0+0+0+0+0+0+voltage\_corret+0+0+0+0+0+0+0.0+dynami...c\_correct+0+0+0+0+0+0+0+0+0+

- `app_weld_set_interface_eip_m2r_other(opt1=[0,0,0,0,0,0,0], opt2=[0,0,0,0,0,0,0], opt3=[0,0,0,0,0,0,0], opt4=[0,0,0,0,0,0,0], opt5=[0,0,0,0,0,0,0], opt6=[0,0,0,0,0,0,0], opt7=[0,0,0,0,0,0,0],...)(p. 529)`

### **12.3.7 app\_weld\_set\_interface\_eip\_r2m\_option(opt1=[0,0,0,0,0,0,0,0], opt2=[0,0,0,0,0,0,0,0], opt3=[0,0,0,0,0,0,0,0], opt4=[0,0,0,0,0,0,0,0], opt5=[0,0,0,0,0,0,0,0], opt6=[0,0,0,0,0,0,0,0],...)**

#### 기능

EtherNet/IP 통신을 지원하는 용접기를 사용하기 위한 통신 인터페이스 설정을 합니다. 로봇제어기에서 용접기로 보내는 통신 데이터 중, 기본으로 제공하는 설정 항목들 (`app_weld_set_interface_eip_r2m_process()`, `app_weld_set_interface_eip_r2m_mode()`, `app_weld_set_interface_eip_r2m_test()`, `app_weld_set_interface_eip_r2m_condition()`) 이외에 필요한 기능들을 해당 명령어를 통해 추가적으로 설정할 수 있습니다. 아래 설정값과 관련된 세부 사항은 해당 용접기의 통신 시그널 데이터시트를 참고하여 기입하기 바랍니다.

#### **❶ 알아두기**

1. EtherNet/IP 통신방식 용접기를 이용한 정상적인 용접기능 수행을 위해서는 8종의 인터페이스 설정 명령어의 설정이 모두 완료되어야 합니다. `app_weld_set_interface_eip_r2m_process()`, `app_weld_set_interface_eip_r2m_mode()`, `app_weld_set_interface_eip_r2m_test()`, `app_weld_set_interface_eip_r2m_condition()`, `app_weld_set_interface_eip_r2m_option()`, `app_weld_set_interface_eip_m2r_process()`, `app_weld_set_interface_eip_m2r_monitoring()`, `app_weld_set_interface_eip_m2r_other()`

#### 인수

인수명	자료형	기본값	설명
opt1	아래참조	아래참조	옵션 항목 (용접기별사양)
opt2			
opt3			
opt4			
opt5			
opt6			

인수명	자료형	기본값	설명
opt7			
opt8			
opt9			
opt10			
opt11			
opt12			
opt13			
opt14			
opt15			

인수 자료형, 기본값, 설명은 아래와 동일

인수명	자료형	기본값	설명
	list(int[7])	0	미사용: 0 사용: 1
		0	데이터 타입(on/off: 0, 선택: 1, 값: 2)
		0	데이터 자릿수(1: 0, 0.1: 1, 0.01: 2)
		0	통신 데이터 위치(byte): 1~255
		0	통신 데이터 위치(bit): 1~255

인수명	자료형	기본값	설명
		0	데이터 사이즈 1-bit(disable Low): 0 1-bit(disable High): 1 2-bit: 2 4-bit: 3 8-bit(byte): 4 15-bit: 5 16-bit(short): 6 32-bit(int): 7
		0	유효한 데이터 사이즈 값(bit)
list(floa t[2])	0	0	데이터 최소값
		0	데이터 최대값

### ❶ 알아두기

데이터 타입(0~2)별 인터페이스 설정 예시는 `app_weld_set_interface_eip_r2m_process()` 절을 참고하세요.

### 리턴

값	설명
0	성공
음수	실패

### 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시

예외	설명
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

예제

```
1 app_weld_set_interface_eip_r2m_option(opt1=[0,0,0,0,0,0,0,0,0,0], opt2=[0,0,0,  
    , opt3=[0,0,0,0,0,0,0,0,0], opt4=[0,0,0,0,0,0,0,0,0], opt5=[0,0,0,0,0,0,0,0,  
    , opt6=[0,0,0,0,0,0,0,0,0], opt7=[0,0,0,0,0,0,0,0,0], opt8=[0,0,0,0,0,0,0,0,  
    , opt9=[0,0,0,0,0,0,0,0,0], opt10=[0,0,0,0,0,0,0,0,0], opt11=[0,0,0,0,0,0,0,0,  
    , opt12=[0,0,0,0,0,0,0,0,0], opt13=[0,0,0,0,0,0,0,0,0], opt14=[0,0,0,0,0,0,0,0,  
    , opt15=[0,0,0,0,0,0,0,0,0])
```

## 관련 명령어

- app\_weld\_set\_interface\_eip\_r2m\_process(welding\_start=[0,0,0,0,0,0,0,0], robot\_ready=[0,0,0,0,0,0,0,0], error\_reset=[0,0,0,0,0,0,0,0])(p. 502)
  - app\_weld\_set\_interface\_eip\_r2m\_mode(welding\_mode=[0,0,0,0,0,0,0,0], s\_2t=[0,0,0,0,0,0,0,0], pulse\_mode=[0,0,0,0,0,0,0,0],wm\_opt1=[0,0,0,0,0,0,0,0])(p. 507)
  - app\_weld\_set\_interface\_eip\_r2m\_test(gas\_test=[0,0,0,0,0,0,0,0], inching\_plus=[0,0,0,0,0,0,0,0], inching\_minus=[0,0,0,0,0,0,0,0], blow\_out\_torch=[0,0,0,0,0,0,0,0], simulation=[0,0,0,0,0,0,0,0], ts\_opt1=[0,0,0,0,0,0,0,0], ...)(p. 511)
  - app\_weld\_set\_interface\_eip\_r2m\_condition(job\_num=[0,0,0,0,0,0,0,0], synergic\_id=[0,0,0,0,0,0,0,0], r\_wire\_feed\_speed=[0,0,0,0,0,0,0,0], voltage\_corret=[0,0,0,0,0,0,0,0], dynamic\_correct=[0,0,0,0,0,0,0,0])(p. 514)
  - app\_weld\_set\_interface\_eip\_r2m\_option(opt1=[0,0,0,0,0,0,0,0], opt2=[0,0,0,0,0,0,0,0], opt3=[0,0,0,0,0,0,0,0], opt4=[0,0,0,0,0,0,0,0], opt5=[0,0,0,0,0,0,0,0], opt6=[0,0,0,0,0,0,0,0], ...)<sup>36</sup>
  - app\_weld\_set\_interface\_eip\_m2r\_process(current\_flow=[0,0,0,0,0,0,0,0], process\_active=[0,0,0,0,0,0,0,0], main\_current=[0,0,0,0,0,0,0,0], machine\_ready=[0,0,0,0,0,0,0,0], comm\_ready=[0,0,0,0,0,0,0,0])(p. 522)
  - app\_weld\_set\_interface\_eip\_m2r\_monitoring(welding\_voltage=[0,0,0,0,0,0,0,0], welding\_current=[0,0,0,0,0,0,0,0], wire\_feed\_speed=[0,0,0,0,0,0,0,0], wire\_stick=[0,0,0,0,0,0,0,0], error=[0,0,0,0,0,0,0,0] ...)(p. 525)
  - app\_weld\_set\_interface\_eip\_m2r\_other(opt1=[0,0,0,0,0,0,0,0], opt2=[0,0,0,0,0,0,0,0], opt3=[0,0,0,0,0,0,0,0], opt4=[0,0,0,0,0,0,0,0], opt5=[0,0,0,0,0,0,0,0], opt6=[0,0,0,0,0,0,0,0], opt7=[0,0,0,0,0,0,0,0], ...)(p. 529)

[36https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/5015183/](https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/5015183/)

### 12.3.8 app\_weld\_set\_interface\_eip\_m2r\_process(**current\_flow=[0,0,0,0,0,0,0,0]**, **process\_active=[0,0,0,0,0,0,0,0]**, **main\_current=[0,0,0,0,0,0,0,0]**, **machine\_ready=[0,0,0,0,0,0,0,0]**, **comm\_ready=[0,0,0,0,0,0,0,0]**)

#### 기능

EtherNet/IP 통신을 지원하는 용접기를 사용하기 위한 통신 인터페이스 설정을 합니다. 용접기에서 로봇제어기로 보내는 통신 데이터 중, 용접 수행을 위한 제어기와 용접기간의 연동 신호의 인터페이스를 설정할 수 있습니다. 아래 설정값과 관련된 세부 사항은 해당 용접기의 통신 시그널 데이터시트를 참고하여 기입하기 바랍니다.

#### 알아두기

1. EtherNet/IP 통신방식 용접기를 이용한 정상적인 용접기능 수행을 위해서는 8종의 인터페이스 설정 명령어의 설정이 모두 완료되어야 합니다.  
`app_weld_set_interface_eip_r2m_process()`, `app_weld_set_interface_eip_r2m_mode()`,  
`app_weld_set_interface_eip_r2m_test()`, `app_weld_set_interface_eip_r2m_condition()`,  
`app_weld_set_interface_eip_r2m_option()`, `app_weld_set_interface_eip_m2r_process()`,  
`app_weld_set_interface_eip_m2r_monitoring()`, `app_weld_set_interface_eip_m2r_other()`
2. 로봇 모션 시작은 용접기로 부터의 `current_flow` 신호와 연동이 되나 `main_current` 항목 설정 시 해당 신호와 연동이 됩니다.
3. 로봇 모션 종료는 용접기로 부터의 `current_flow` 신호와 연동이 되나 `process_active` 항목 설정 시 해당 신호와 연동이 됩니다.

#### 인수

인수명	자료형	기본값	설명
<code>current_flow</code>	아래참고	아래참고	용접전류 발생중 (용접기별사양)
<code>process_active</code>			용접공정 활성화 (용접기별사양)
<code>main_current</code>			본용접전류 발생중 (용접기별사양)
<code>machine_ready</code>			용접 대기중 (용접기별사양)
<code>comm_ready</code>			통신 대기중 (용접기별사양)

인수 자료형, 기본값, 설명은 아래와 동일

인수명	자료형	기본값	설명
	list(int[7])	0	미사용: 0 사용: 1
		0	데이터 타입(on/off: 0, 선택: 1, 값: 2)
		0	데이터 자릿수(1: 0, 0.1: 1, 0.01: 2)
		0	통신 데이터 위치 (byte): 1~255
		0	통신 데이터 위치 (bit): 1~255
		0	데이터 사이즈 1-bit(disable Low): 0 1-bit(disable High): 1 2-bit: 2 4-bit: 3 8-bit(byte): 4 15-bit: 5 16-bit(short): 6 32-bit(int): 7
		0	유효한 데이터 사이즈 값(bit)
		0	데이터 최소값
		0	데이터 최대값

**i 알아두기**

데이터 타입(0~2)별 인터페이스 설정 예시는 `app_weld_set_interface_eip_r2m_process()` 절을 참고하세요.

## 리턴

값	설명
0	성공
음수	실패

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

1	app_weld_set_interface_eip_m2r_process(current_flow=[1,0,0,0,0,0,0,1,0,0], process_active=[1,0,0,0,6,0,1,0,0], main_current=[1,0,0,0,5,0,1,0,0], machine_ready=[0,0,0,0,0,0,0,0,0], comm_ready=[0,0,0,0,0,0,0,0])
---	---

## 관련 명령어

- [app\\_weld\\_set\\_interface\\_eip\\_r2m\\_process\(welding\\_start=\[0,0,0,0,0,0,0,0\], robot\\_ready=\[0,0,0,0,0,0,0,0\], error\\_reset=\[0,0,0,0,0,0,0,0\]\)\(p. 502\)](#)
- [app\\_weld\\_set\\_interface\\_eip\\_r2m\\_mode\(welding\\_mode=\[0,0,0,0,0,0,0,0\], s\\_2t=\[0,0,0,0,0,0,0,0\], pulse\\_mode=\[0,0,0,0,0,0,0,0\],wm\\_opt1=\[0,0,0,0,0,0,0,0\]\)\(p. 507\)](#)
- [app\\_weld\\_set\\_interface\\_eip\\_r2m\\_test\(gas\\_test=\[0,0,0,0,0,0,0,0\], inching\\_plus=\[0,0,0,0,0,0,0,0\], inching\\_minus=\[0,0,0,0,0,0,0,0\], blow\\_out\\_torch=\[0,0,0,0,0,0,0,0\], simulation=\[0,0,0,0,0,0,0,0\], ts\\_opt1=\[0,0,0,0,0,0,0,0\], ...\)\(p. 511\)](#)

- `app_weld_set_interface_eip_r2m_condition(job_num=[0,0,0,0,0,0,0,0], synergic_id=[0,0,0,0,0,0,0,0], r_wire_feed_speed=[0,0,0,0,0,0,0,0], voltage_corret=[0,0,0,0,0,0,0,0], dynamic_correct=[0,0,0,0,0,0,0,0])(p. 514)`
- `app_weld_set_interface_eip_r2m_option(opt1=[0,0,0,0,0,0,0,0], opt2=[0,0,0,0,0,0,0,0], opt3=[0,0,0,0,0,0,0,0], opt4=[0,0,0,0,0,0,0,0], opt5=[0,0,0,0,0,0,0,0], opt6=[0,0,0,0,0,0,0,0], ...)(p. 518)`
- `app_weld_set_interface_eip_m2r_process(current_flow=[0,0,0,0,0,0,0,0], process_active=[0,0,0,0,0,0,0,0], main_current=[0,0,0,0,0,0,0,0], machine_ready=[0,0,0,0,0,0,0,0], comm_ready=[0,0,0,0,0,0,0,0])37`
- `app_weld_set_interface_eip_m2r_monitoring(welding_voltage=[0,0,0,0,0,0,0,0], welding_current=[0,0,0,0,0,0,0,0], wire_feed_speed=[0,0,0,0,0,0,0,0], wire_stick=[0,0,0,0,0,0,0,0], error=[0,0,0,0,0,0,0,0] ...)(p. 525)`
- `app_weld_set_interface_eip_m2r_other(opt1=[0,0,0,0,0,0,0,0], opt2=[0,0,0,0,0,0,0,0], opt3=[0,0,0,0,0,0,0,0], opt4=[0,0,0,0,0,0,0,0], opt5=[0,0,0,0,0,0,0,0], opt6=[0,0,0,0,0,0,0,0], opt7=[0,0,0,0,0,0,0,0], ...)(p. 529)`

### **12.3.9 app\_weld\_set\_interface\_eip\_m2r\_monitoring(welding\_voltage=[0,0,0,0,0,0,0,0], welding\_current=[0,0,0,0,0,0,0,0], wire\_feed\_speed=[0,0,0,0,0,0,0,0], wire\_stick=[0,0,0,0,0,0,0,0], error=[0,0,0,0,0,0,0,0] ...)**

#### 기능

EtherNet/IP 통신을 지원하는 용접기를 사용하기 위한 통신 인터페이스 설정을 합니다. 용접기에서 로봇제어기로 보내는 통신 데이터 중, 용접기 상태 모니터링과 관련한 인터페이스를 설정합니다. 아래 설정값과 관련된 세부 사항은 해당 용접기의 통신 시그널 데이터시트를 참고하여 기입하기 바랍니다

#### **❶ 알아두기**

EtherNet/IP 통신방식 용접기를 이용한 정상적인 용접기능 수행을 위해서는 8종의 인터페이스 설정 명령어의 설정이 모두 완료되어야 합니다.

```
app_weld_set_interface_eip_r2m_process(), app_weld_set_interface_eip_r2m_mode(),
app_weld_set_interface_eip_r2m_test(), app_weld_set_interface_eip_r2m_condition(),
app_weld_set_interface_eip_r2m_option(), app_weld_set_interface_eip_m2r_process(),
app_weld_set_interface_eip_m2r_monitoring(), app_weld_set_interface_eip_m2r_other()
```

<sup>37</sup><https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/5015187/>

app\_weld\_set\_interface\_eip\_m2r\_process+current\_flow+0+0+0+0+0+0+0+0+process\_active+0+0+0+0+0+0+0+main\_current+0+0+0+0+0+0+0+0+machine\_ready+0+0+0+0+0+0+0+0+comm\_ready+0+0+0+0+0+0+0+0+0+0

## 인수

인수명	자료형	기본값	설명
welding_voltage	아래참고	아래참고	실제 용접 전압 (용접기별사양)
welding_current			실제 용접 전류 (용접기별사양)
wire_feed_speed			실제 와이어 피딩 속도 (용접기별사양)
wire_stick			와이어 팁 웅착 상황 여부 (용접기별사양)
error			에러 상황 여부 (용접기별사양)
error_num			에러 번호 (용접기별사양)

인수 자료형, 기본값, 설명은 아래와 동일

인수명	자료형	기본값	설명
	list(int[7])	0	미사용: 0 사용: 1
		0	데이터 타입(on/off: 0, 선택: 1, 값: 2)
		0	데이터 자릿수(1: 0, 0.1: 1, 0.01: 2)
		0	통신 데이터 위치(byte): 1~255
		0	통신 데이터 위치(bit): 1~255

인수명	자료형	기본값	설명
		0	데이터 사이즈 1-bit(disable Low): 0 1-bit(disable High): 1 2-bit: 2 4-bit: 3 8-bit(byte): 4 15-bit: 5 16-bit(short): 6 32-bit(int): 7
		0	유효한 데이터 사이즈 값(bit)
	list(float[2])	0	데이터 최소값
		0	데이터 최대값

### ● 알아두기

데이터 타입(0~2)별 인터페이스 설정 예시는 `app_weld_set_interface_eip_r2m_process()` 절을 참고하세요

## 리턴

값	설명
0	성공
음수	실패

## 예외

예외	설명
<code>DR_Error</code> ( <code>DR_ERROR_TYPE</code> )	인수들의 데이터형 오류 시
<code>DR_Error</code> ( <code>DR_ERROR_VALUE</code> )	인수의 값이 유효하지 않을 시

예외	설명
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```
1 app_weld_set_interface_eip_m2r_monitoring(welding_voltage=[0,0,0,0,0,0,0,0,0,0], welding_current=[0,0,0,0,0,0,0,0,0,0], wire_feed_speed=[0,0,0,0,0,0,0,0,0,0], wire_stick=[0,0,0,0,0,0,0,0,0,0], error=[0,0,0,0,0,0,0,0,0,0], error_num=[0,0,0,0,0,0,0,0,0,0])
```

## 관련 명령어

- app\_weld\_set\_interface\_eip\_r2m\_process(welding\_start=[0,0,0,0,0,0,0,0], robot\_ready=[0,0,0,0,0,0,0,0], error\_reset=[0,0,0,0,0,0,0,0])(p. 502)
  - app\_weld\_set\_interface\_eip\_r2m\_mode(welding\_mode=[0,0,0,0,0,0,0,0], s\_2t=[0,0,0,0,0,0,0,0], pulse\_mode=[0,0,0,0,0,0,0,0],wm\_opt1=[0,0,0,0,0,0,0,0])(p. 507)
  - app\_weld\_set\_interface\_eip\_r2m\_test(gas\_test=[0,0,0,0,0,0,0,0], inching\_plus=[0,0,0,0,0,0,0,0], inching\_minus=[0,0,0,0,0,0,0,0], blow\_out\_torch=[0,0,0,0,0,0,0,0], simulation=[0,0,0,0,0,0,0,0], ts\_opt1=[0,0,0,0,0,0,0,0], ...)(p. 511)
  - app\_weld\_set\_interface\_eip\_r2m\_condition(job\_num=[0,0,0,0,0,0,0,0], synergic\_id=[0,0,0,0,0,0,0,0], r\_wire\_feed\_speed=[0,0,0,0,0,0,0,0], voltage\_corret=[0,0,0,0,0,0,0,0], dynamic\_correct=[0,0,0,0,0,0,0,0])(p. 514)
  - app\_weld\_set\_interface\_eip\_r2m\_option(opt1=[0,0,0,0,0,0,0,0], opt2=[0,0,0,0,0,0,0,0], opt3=[0,0,0,0,0,0,0,0], opt4=[0,0,0,0,0,0,0,0], opt5=[0,0,0,0,0,0,0,0], opt6=[0,0,0,0,0,0,0,0], ...)(p. 518)
  - app\_weld\_set\_interface\_eip\_m2r\_process(current\_flow=[0,0,0,0,0,0,0,0], process\_active=[0,0,0,0,0,0,0,0], main\_current=[0,0,0,0,0,0,0,0], machine\_ready=[0,0,0,0,0,0,0,0], comm\_ready=[0,0,0,0,0,0,0,0])(p. 522)
  - app\_weld\_set\_interface\_eip\_m2r\_monitoring(welding\_voltage=[0,0,0,0,0,0,0,0], welding\_current=[0,0,0,0,0,0,0,0], wire\_feed\_speed=[0,0,0,0,0,0,0,0], wire\_stick=[0,0,0,0,0,0,0,0], error=[0,0,0,0,0,0,0,0] ...)<sup>38</sup>
  - app\_weld\_set\_interface\_eip\_m2r\_other(opt1=[0,0,0,0,0,0,0,0], opt2=[0,0,0,0,0,0,0,0], opt3=[0,0,0,0,0,0,0,0], opt4=[0,0,0,0,0,0,0,0], opt5=[0,0,0,0,0,0,0,0], opt6=[0,0,0,0,0,0,0,0], opt7=[0,0,0,0,0,0,0,0], ...)(p. 529)

[38https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/5015191/](https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/5015191/)

### 12.3.10 app\_weld\_set\_interface\_eip\_m2r\_other(opt1=[0,0,0,0,0,0,0,0,0], opt2=[0,0,0,0,0,0,0,0,0], opt3=[0,0,0,0,0,0,0,0,0], opt4=[0,0,0,0,0,0,0,0,0], opt5=[0,0,0,0,0,0,0,0,0], opt6=[0,0,0,0,0,0,0,0,0], opt7=[0,0,0,0,0,0,0,0,0],...)

#### 기능

EtherNet/IP 통신을 지원하는 용접기를 사용하기 위한 통신 인터페이스 설정을 합니다. 용접기에서 로봇제어기로 보내는 통신 데이터 중, 기본으로 제공하는 설정 항목들 (app\_weld\_set\_interface\_eip\_m2r\_process(), app\_weld\_set\_interface\_eip\_m2r\_monitoring(), app\_weld\_set\_interface\_eip\_m2r\_other()) 이외에 필요한 기능의 인터페이스 설정을 해당 명령어를 통해 추가적으로 수행할 수 있습니다. 아래 설정값과 관련된 세부 사항은 해당 용접기의 통신 시그널 데이터시트를 참고하여 기입하기 바랍니다

#### 알아두기

EtherNet/IP 통신방식 용접기를 이용한 정상적인 용접기능 수행을 위해서는 8종의 인터페이스 설정 명령어의 설정이 모두 완료되어야 합니다. app\_weld\_set\_interface\_eip\_r2m\_process(), app\_weld\_set\_interface\_eip\_r2m\_mode(), app\_weld\_set\_interface\_eip\_r2m\_test(), app\_weld\_set\_interface\_eip\_r2m\_condition(), app\_weld\_set\_interface\_eip\_r2m\_option(), app\_weld\_set\_interface\_eip\_m2r\_process(), app\_weld\_set\_interface\_eip\_m2r\_monitoring(), app\_weld\_set\_interface\_eip\_m2r\_other()

#### 인수

인수명	자료형	기본값	설명
opt1	사양정보		
opt2	사양정보		
opt3	사양정보		
opt4	사양정보		

인수명	자료형	기본값	설명
opt5	사양정보		
opt6	사양정보		
opt7	사양정보		
opt8	사양정보		
opt9	사양정보		
opt10	사양정보		

인수 자료형, 기본값, 설명은 아래와 동일

인수명	자료형	기본값	설명
	list(int[7])	0	미사용: 0 사용: 1
		0	데이터 타입(on/off: 0, 선택: 1, 값: 2)
		0	데이터 자릿수(1: 0, 0.1: 1, 0.01: 2)
		0	통신 데이터 위치 (byte): 1~255
		0	통신 데이터 위치 (bit): 1~255

인수명	자료형	기본값	설명
		0	데이터 사이즈 1-bit(disable Low): 0 1-bit(disable High): 1 2-bit: 2 4-bit: 3 8-bit(byte): 4 15-bit: 5 16-bit(short): 6 32-bit(int): 7
		0	유효한 데이터 사이즈 값(bit)
	list(float[2])	0	데이터 최소값
		0	데이터 최대값

### ❶ 알아두기

데이터 타입(0~2)별 인터페이스 설정 예시는 `app_weld_set_interface_eip_r2m_process()` 절을 참고하세요.

### 리턴

값	설명
0	성공
음수	실패

### 예외

예외	설명
<code>DR_Error</code> ( <code>DR_ERROR_TYPE</code> )	인수들의 데이터형 오류 시

예외	설명
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

1	app_weld_set_interface_eip_m2r_other(opt1=[1,2,1,12,0,6,15,0.0,25.5], opt2=[1,0,0,0,1,0,1,0,0], opt3=[0,0,0,0,0,0,0,0,0], opt4=[0,0,0,0,0,0,0,0,0] , opt5=[0,0,0,0,0,0,0,0,0], opt6=[0,0,0,0,0,0,0,0,0], opt7=[0,0,0,0,0,0,0,0,0] , opt8=[0,0,0,0,0,0,0,0,0], opt9=[0,0,0,0,0,0,0,0,0], opt10=[0,0,0,0,0,0,0,0,0])
---	--

## 관련 명령어

- app\_weld\_set\_interface\_eip\_r2m\_process(welding\_start=[0,0,0,0,0,0,0,0], robot\_ready=[0,0,0,0,0,0,0,0], error\_reset=[0,0,0,0,0,0,0,0])(p. 502)
- app\_weld\_set\_interface\_eip\_r2m\_mode(welding\_mode=[0,0,0,0,0,0,0,0], s\_2t=[0,0,0,0,0,0,0,0], pulse\_mode=[0,0,0,0,0,0,0,0],wm\_opt1=[0,0,0,0,0,0,0,0])(p. 507)
- app\_weld\_set\_interface\_eip\_r2m\_test(gas\_test=[0,0,0,0,0,0,0,0], inching\_plus=[0,0,0,0,0,0,0,0], inching\_minus=[0,0,0,0,0,0,0,0], blow\_out\_torch=[0,0,0,0,0,0,0,0], simulation=[0,0,0,0,0,0,0,0], ts\_opt1=[0,0,0,0,0,0,0,0], ...)(p. 511)
- app\_weld\_set\_interface\_eip\_r2m\_condition(job\_num=[0,0,0,0,0,0,0,0], synergic\_id=[0,0,0,0,0,0,0,0], r\_wire\_feed\_speed=[0,0,0,0,0,0,0,0], voltage\_corret=[0,0,0,0,0,0,0,0.0], dynamic\_correct=[0,0,0,0,0,0,0,0])(p. 514)
- app\_weld\_set\_interface\_eip\_r2m\_option(opt1=[0,0,0,0,0,0,0,0], opt2=[0,0,0,0,0,0,0,0], opt3=[0,0,0,0,0,0,0,0], opt4=[0,0,0,0,0,0,0,0], opt5=[0,0,0,0,0,0,0,0], opt6=[0,0,0,0,0,0,0,0], ...)(p. 518)
- app\_weld\_set\_interface\_eip\_m2r\_process(current\_flow=[0,0,0,0,0,0,0,0], process\_active=[0,0,0,0,0,0,0,0], main\_current=[0,0,0,0,0,0,0,0], machine\_ready=[0,0,0,0,0,0,0,0], comm\_ready=[0,0,0,0,0,0,0,0])(p. 522)
- app\_weld\_set\_interface\_eip\_m2r\_monitoring(welding\_voltage=[0,0,0,0,0,0,0,0], welding\_current=[0,0,0,0,0,0,0,0], wire\_feed\_speed=[0,0,0,0,0,0,0,0], wire\_stick=[0,0,0,0,0,0,0,0], error=[0,0,0,0,0,0,0,0] ...)(p. 525)

- app\_weld\_set\_interface\_eip\_m2r\_other(opt1=[0,0,0,0,0,0,0], opt2=[0,0,0,0,0,0,0],  
 opt3=[0,0,0,0,0,0,0], opt4=[0,0,0,0,0,0,0], opt5=[0,0,0,0,0,0,0], opt6=[0,0,0,0,0,0,0],  
 opt7=[0,0,0,0,0,0,0],...)<sup>39</sup>

**12.3.11 app\_weld\_set\_weld\_cond\_digital(flag\_dry\_run=0, vel\_target=0, vel\_min=0, vel\_max=0, welding\_mode=0, s\_2t=0, pulse\_mode=0, wm\_opt1=0, simulation=0, ts\_opt1=0, ts\_opt2=0,...)**

기능

통신방식 용접조건을 설정합니다. 용접조건은 용접기능 활성화 (app\_weld\_enable\_digital()) 부터 비활성화(app\_weld\_disable\_digital())로 정의한 용접구간 내에서만 유효하며 이외에 실행 시 에러가 발생합니다. 용접조건으로 설정할 수 있는 항목은 다음 명령어(app\_weld\_set\_interface\_eip\_r2m\_mode(), app\_weld\_set\_interface\_eip\_r2m\_condition(), app\_weld\_set\_interface\_eip\_r2m\_option())에 해당하는 용접기와 통신 인터페이스 설정을 완료한 항목에 대해서만 가능합니다.

하나의 용접구간 내에서는 하나의 용접조건만 허용되며 용접 중 app\_weld\_adj\_welding\_cond\_digital() 명령어를 통해 용접조건을 조정하거나, 티치펜던트의 용접조건 조정 popup에서 전압조정/동적인자조정/피딩속도/속도(및 위빙옵션)를 조정할 수 있습니다. 다만 티치펜던트에서의 용접조건 조정은 명령어를 통한 용접조건 조정상태가 RESET 상태(즉, app\_weld\_set\_weld\_cond\_digital())으로 지정한 용접조건 설정)인 경우에만 가능합니다.

## 알아두기

1. 전압조정(voltage correction) : 아크 길이를 조정 합니다.
  2. 동적인자조정(dynamic correction) : 아크 특성을 조정 합니다.

인수

인수명	자료형	기본값	설명
flag_dry_run	int	0	모의용접(Dry-run) 모드 실용접(0) 모의용접(1) : 모션/위빙/옵셋만 진행
vel_target	float	0	목표속도 (mm/sec) * 티치펜던트의 입력단위와 다른 것에 유의할 것(Cm/min)

[39https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/5015195/](https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/5015195/)

인수명	자료형	기본값	설명
vel_min	float	0	목표속도 조정 최소값 (mm/sec) * 티치펜던트의 입력단위와 다른 것에 유의할 것(Cm/min)
vel_max	float	0	목표속도 조정 최대값 (mm/sec) * 티치펜던트의 입력단위와 다른 것에 유의할 것(Cm/min)
welding_mode	Int	0	용접 모드 설정
s_2t	Int	0	2T, 2T special 설정
pulse_mode	Int	0	펄스 모드 설정
wm_opt1	Int	0	용접모드 옵션1 설정
simulation	int	0	시뮬레이션 모드 설정
ts_opt1	Int	0	테스트 신호 옵션1 설정
ts_opt2	Int	0	테스트 신호 옵션2 설정
Job_num	Int	0	Job 번호 설정
synergic_id	Int	0	Synergic ID 설정
r_wire_feed_speed	float	0	와이어 피딩 속도 설정
voltage_correct	float	0	전압 조정 설정
dynamic_correct	float	0	동적 인자 조정 설정
r_opt1	float	0	옵션1 설정
r_opt2	float	0	옵션2 설정
r_opt3	float	0	옵션3 설정
r_opt4	float	0	옵션4 설정
r_opt5	float	0	옵션5 설정

인수명	자료형	기본값	설명
r_opt6	float	0	옵션6 설정
r_opt7	float	0	옵션7 설정
r_opt8	float	0	옵션8 설정
r_opt9	float	0	옵션9 설정
r_opt10	float	0	옵션10 설정
r_opt11	float	0	옵션11 설정
r_opt12	float	0	옵션12 설정
r_opt13	float	0	옵션13 설정
r_opt14	float	0	옵션14 설정
r_opt15	float	0	옵션15 설정

### 리턴

값	설명
0	설정 성공
음수값	설정 실패

### 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시

예외	설명
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 app_weld_enable_digital()
2 app_weld_set_weld_cond_digital(flag_dry_run=0, vel_target=16, vel_min=0.00,
   vel_max=16.67, welding_mode=3, s_2t=0, pulse_mode=0, wm_opt1=0,
   simulation=0, ts_opt1=0, ts_opt2=0, job_num=4, synergic_id=0,
   r_wire_feed_speed=10, voltage_correct=0, dynamic_correct=0, r_opt1=0,
   r_opt2=0, r_opt3=0, r_opt4=0, r_opt5=0, r_opt6=0, r_opt7=0, r_opt8=0,
   r_opt9=0, r_opt10=0, r_opt11=0, r_opt12=0, r_opt13=0, r_opt14=0, r_opt15=0)
   #용접속도=60mm/sec(=1Cm/min), 용접모드=3, Job번호=4, 와이어 피딩 속도=10m/min
3
4
5 app_weld_disable_digital()

```

## 관련 명령어

- [app\\_weld\\_set\\_interface\\_eip\\_r2m\\_process\(welding\\_start=\[0,0,0,0,0,0,0,0\], robot\\_ready=\[0,0,0,0,0,0,0,0\], error\\_reset=\[0,0,0,0,0,0,0,0\]\)\(p. 502\)](#)
- [app\\_weld\\_set\\_interface\\_eip\\_r2m\\_mode\(welding\\_mode=\[0,0,0,0,0,0,0,0\], s\\_2t=\[0,0,0,0,0,0,0,0\], pulse\\_mode=\[0,0,0,0,0,0,0,0\],wm\\_opt1=\[0,0,0,0,0,0,0,0\]\)\(p. 507\)](#)
- [app\\_weld\\_set\\_interface\\_eip\\_r2m\\_test\(gas\\_test=\[0,0,0,0,0,0,0,0\], inching\\_plus=\[0,0,0,0,0,0,0,0\], inching\\_minus=\[0,0,0,0,0,0,0,0\], blow\\_out\\_torch=\[0,0,0,0,0,0,0,0\], simulation=\[0,0,0,0,0,0,0,0\], ts\\_opt1=\[0,0,0,0,0,0,0,0\], ...\)\(p. 511\)](#)
- [app\\_weld\\_set\\_interface\\_eip\\_r2m\\_condition\(job\\_num=\[0,0,0,0,0,0,0,0\], synergic\\_id=\[0,0,0,0,0,0,0,0\], r\\_wire\\_feed\\_speed=\[0,0,0,0,0,0,0,0\], voltage\\_correct=\[0,0,0,0,0,0,0,0\], dynamic\\_correct=\[0,0,0,0,0,0,0,0\]\)\(p. 514\)](#)
- [app\\_weld\\_set\\_interface\\_eip\\_r2m\\_option\(opt1=\[0,0,0,0,0,0,0,0\], opt2=\[0,0,0,0,0,0,0,0\], opt3=\[0,0,0,0,0,0,0,0\], opt4=\[0,0,0,0,0,0,0,0\], opt5=\[0,0,0,0,0,0,0,0\], opt6=\[0,0,0,0,0,0,0,0\], ...\)\(p. 518\)](#)
- [app\\_weld\\_set\\_interface\\_eip\\_m2r\\_process\(current\\_flow=\[0,0,0,0,0,0,0,0\], process\\_active=\[0,0,0,0,0,0,0,0\], main\\_current=\[0,0,0,0,0,0,0,0\], machine\\_ready=\[0,0,0,0,0,0,0,0\], comm\\_ready=\[0,0,0,0,0,0,0,0\]\)\(p. 522\)](#)
- [app\\_weld\\_set\\_interface\\_eip\\_m2r\\_monitoring\(welding\\_voltage=\[0,0,0,0,0,0,0,0\], welding\\_current=\[0,0,0,0,0,0,0,0\], wire\\_feed\\_speed=\[0,0,0,0,0,0,0,0\], wire\\_stick=\[0,0,0,0,0,0,0,0\], error=\[0,0,0,0,0,0,0,0\] ...\)\(p. 525\)](#)

- `app_weld_set_interface_eip_m2r_other(opt1=[0,0,0,0,0,0,0], opt2=[0,0,0,0,0,0,0], opt3=[0,0,0,0,0,0,0], opt4=[0,0,0,0,0,0,0], opt5=[0,0,0,0,0,0,0], opt6=[0,0,0,0,0,0,0], opt7=[0,0,0,0,0,0,0],...)(p. 529)`
- `app_weld_enable_digital()(p. 498)`
- `app_weld_set_weld_cond_digital(flag_dry_run=0, vel_target=0, vel_min=0, vel_max=0, welding_mode=0, s_2t=0, pulse_mode=0, wm_opt1=0, simulation=0, ts_opt1=0, ts_opt2=0,...)40`
- `app_weld_adj_welding_cond_digital(flag_reset=None, f_target=None, vel_target=None, wv_offset=None, wv_width_ratio=None, dynamic_cor=None, voltage_cor=None, job_number=None, synergic_id=None)(p. 537)`
- `app_weld_adj_welding_cond_digital(flag_reset=None, f_target=None, vel_target=None, wv_offset=None, wv_width_ratio=None, dynamic_cor=None, voltage_cor=None, job_number=None, synergic_id=None)(p. 537)`
- `app_weld_disable_digital()(p. 500)`
- `app_weld_weave_cond_trapezoidal(wv_offset=[0,0], wv_ang=0, wv_param=[0,1.5,0,-1.5,0.3,0.1,0.3,0.3,0.1,0.3])(p. 560)`
- `app_weld_weave_cond_zigzag(wv_offset=[0,0], wv_ang=0, wv_param=[3,0.6])(p. 562)`
- `app_weld_weave_cond_circular(wv_offset=[0,0], wv_ang=0, wv_param=[3,3,0.3,0.3])(p. 564)`
- `app_weld_weave_cond_sinusoidal(wv_offset=[0,0], wv_ang=0, wv_param=[3,0.6])(p. 566)`

### 12.3.12 `app_weld_adj_welding_cond_digital(flag_reset=None, f_target=None, vel_target=None, wv_offset=None, wv_width_ratio=None, dynamic_cor=None, voltage_cor=None, job_number=None, synergic_id=None)`

#### 기능

통신방식 용접기를 이용한 용접 중 용접 및 위빙조건을 조정합니다. 일반적으로 연속된 경로에서 구간별로 용접조건을 변경하고자 할 때 모션명령어(`moveL()`, `moveC()`, `moveB()`, `moveSx()`) 호출 직전에 사용합니다. 본 명령어로 조정 인자를 입력한 경우 해당하는 용접 및 위빙조건이 조정되며 이 때에는 TP의 용접모니터링 정보창에서 용접/위빙조건을 실시간으로 조정할 수 없습니다. 조정조건에서 본조건(`app_weld_set_weld_cond_digital()` 및 `app_weld_weave_cond_trapezoidal()` 등으로 설정한 용접/위빙조건)으로 복귀하려면 `flag_reset=1`로 실행하세요. `flag_reset=1` 설정 시 TP에서 실시간으로 조정한 최종 조건으로 복귀되며(실시간으로 조정불가능한 위빙폭의 비(`wv_width_ratio`)는 1로 변경됩니다.) TP에서 용접조건을 실시간으로 조정할 수 있습니다.

<sup>40</sup><https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/5015199/>

`app_weld_set_weld_cond_digital+flag_dry_run+0+vel_target+0+vel_min+0+vel_max+0+welding_mode+0+s_2t+0+pulse_mode+0+wm_opt1+0+simulation+0+ts_opt1+0+ts_opt2+0+...`

## 인수

인수명	자료형	기본값	설명
flag_reset	int	0	0: 조정값 적용 1: 기준목표(app_weld_set_weld_cond_digital()) 값 적용
f_target	float	-	피딩속도 (m/min)
vel_target	float	-	목표속도 (mm/sec) * 티치펜던트의 입력단위와 다른 것에 유의할 것(Cm/min)
wv_offset	float[2]	-	위빙좌표계-y방향 옵셋 (mm)
		-	위빙좌표계-z방향 옵셋 (mm)
wv_width_ratio	float	-	변경위빙폭/설정위빙폭 의 비 (0~2)
dynamic_cor	float	-	동적 인자 조정
voltage_cor	float	-	전압 조정
job_number	float	-	Job 번호
synergic_id	float	-	Synergic ID

### ● 알아두기

인수 vel\_target/wv\_offset/wv\_width\_ratio/dynamic\_cor/voltage\_cor/job\_number/synergic\_id 중 값을 지정하지 않는 조건은 현재의 조건(실시간으로 조정한 조건 포함)이 유지되므로 조정을 원하는 인수만 설정하세요. 단 wv\_offset의 경우 y방향 또는 z방향 중 하나만 조정하더라도 배열의 두 값을 모두 입력하여야 합니다.

## 리턴

값	설명
0	설정 성공
음수값	설정 실패

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 movej(posj(0,0,90,0,90,0),v=30,a=60)
2
3 pt1= posx(559, 434.5, 651.5, 45, 180, 45)
4 pt2= posx(559, 434.5, 151.5, 45, 180, 45)
5 pt3= posx(559, 0.0, 151.5, 45, 180, 45)
6
7 app_weld_enable_digital()
8
9 app_weld_set_weld_cond_digital(flag_dry_run=0, vel_target=16, vel_min=0.00,
10   vel_max=16.67, welding_mode=3, s_2t=0, pulse_mode=0, wm_opt1=0,
11   simulation=0, ts_opt1=0, ts_opt2=0, job_num=4, synergic_id=0,
12   r_wire_feed_speed=15, voltage_correct=0, dynamic_correct=0, r_opt1=0,
13   r_opt2=0, r_opt3=0, r_opt4=0, r_opt5=0, r_opt6=0, r_opt7=0, r_opt8=0,
14   r_opt9=0, r_opt10=0, r_opt11=0, r_opt12=0, r_opt13=0, r_opt14=0, r_opt15=0)
15
16 movel(pt1, v=5, a=5, r=30, app_type=DR_MV_APP_WELD)
17 app_weld_adj_welding_cond_digital(flag_reset=0, f_target=10, vel_target=16,
18   wv_offset=[20,30], wv_width_ratio=0.5,
19   dynamic_cor=0, voltage_cor=0, job_number=5, synergic_id=4)
20
21 movel(pt2, v=5, a=5, r=30, app_type=DR_MV_APP_WELD)
22 app_weld_adj_welding_cond_digital(flag_reset=1)
23 movel(pt3, v=5, a=5, app_type=DR_MV_APP_WELD)
# 시작위치 → pt1 : 초기설정용접조건 적용 (Job번호:4, Synergic ID:0, 피딩속도: 15m/min)
# pt1 → pt2 : 조정조건 적용 (Job번호:5, Synergic ID:4, 피딩속도: 15m/min)
# pt2 → pt3 : 초기설정 적용 초기설정용접조건 적용 (Job번호:4, Synergic ID:0, 피딩속도: 15m/min)

```

21 app\_weld\_disable\_digital()

## 관련 명령어

- [app\\_weld\\_set\\_interface\\_eip\\_r2m\\_process\(welding\\_start=\[0,0,0,0,0,0,0,0\], robot\\_ready=\[0,0,0,0,0,0,0,0\], error\\_reset=\[0,0,0,0,0,0,0,0\]\)\(p. 502\)](#)
- [app\\_weld\\_set\\_interface\\_eip\\_r2m\\_mode\(welding\\_mode=\[0,0,0,0,0,0,0,0\], s\\_2t=\[0,0,0,0,0,0,0,0\], pulse\\_mode=\[0,0,0,0,0,0,0,0\], wm\\_opt1=\[0,0,0,0,0,0,0,0\]\)\(p. 507\)](#)
- [app\\_weld\\_set\\_interface\\_eip\\_r2m\\_test\(gas\\_test=\[0,0,0,0,0,0,0,0\], inching\\_plus=\[0,0,0,0,0,0,0,0\], inching\\_minus=\[0,0,0,0,0,0,0,0\], blow\\_out\\_torch=\[0,0,0,0,0,0,0,0\], simulation=\[0,0,0,0,0,0,0,0\], ts\\_opt1=\[0,0,0,0,0,0,0,0\], ...\)\(p. 511\)](#)
- [app\\_weld\\_set\\_interface\\_eip\\_r2m\\_condition\(job\\_num=\[0,0,0,0,0,0,0,0\], synergic\\_id=\[0,0,0,0,0,0,0,0\], r\\_wire\\_feed\\_speed=\[0,0,0,0,0,0,0,0\], voltage\\_corret=\[0,0,0,0,0,0,0,0\], dynamic\\_correct=\[0,0,0,0,0,0,0,0\]\)\(p. 514\)](#)
- [app\\_weld\\_set\\_interface\\_eip\\_r2m\\_option\(opt1=\[0,0,0,0,0,0,0,0\], opt2=\[0,0,0,0,0,0,0,0\], opt3=\[0,0,0,0,0,0,0,0\], opt4=\[0,0,0,0,0,0,0,0\], opt5=\[0,0,0,0,0,0,0,0\], opt6=\[0,0,0,0,0,0,0,0\], ...\)\(p. 518\)](#)
- [app\\_weld\\_set\\_interface\\_eip\\_m2r\\_process\(current\\_flow=\[0,0,0,0,0,0,0,0\], process\\_active=\[0,0,0,0,0,0,0,0\], main\\_current=\[0,0,0,0,0,0,0,0\], machine\\_ready=\[0,0,0,0,0,0,0,0\], comm\\_ready=\[0,0,0,0,0,0,0,0\]\)\(p. 522\)](#)
- [app\\_weld\\_set\\_interface\\_eip\\_m2r\\_monitoring\(welding\\_voltage=\[0,0,0,0,0,0,0,0\], welding\\_current=\[0,0,0,0,0,0,0,0\], wire\\_feed\\_speed=\[0,0,0,0,0,0,0,0\], wire\\_stick=\[0,0,0,0,0,0,0,0\], error=\[0,0,0,0,0,0,0,0\] ...\)\(p. 525\)](#)
- [app\\_weld\\_set\\_interface\\_eip\\_m2r\\_other\(opt1=\[0,0,0,0,0,0,0,0\], opt2=\[0,0,0,0,0,0,0,0\], opt3=\[0,0,0,0,0,0,0,0\], opt4=\[0,0,0,0,0,0,0,0\], opt5=\[0,0,0,0,0,0,0,0\], opt6=\[0,0,0,0,0,0,0,0\], opt7=\[0,0,0,0,0,0,0,0\], ...\)\(p. 529\)](#)
- [app\\_weld\\_enable\\_digital\(\)\(p. 498\)](#)
- [app\\_weld\\_adj\\_welding\\_cond\\_digital\(flag\\_reset=None, f\\_target=None, vel\\_target=None, wv\\_offset=None, wv\\_width\\_ratio=None, dynamic\\_cor=None, voltage\\_cor=None, job\\_number=None, synergic\\_id=None\)<sup>41</sup>](#)
- [app\\_weld\\_adj\\_welding\\_cond\\_digital\(flag\\_reset=None, f\\_target=None, vel\\_target=None, wv\\_offset=None, wv\\_width\\_ratio=None, dynamic\\_cor=None, voltage\\_cor=None, job\\_number=None, synergic\\_id=None\)<sup>42</sup>](#)
- [app\\_weld\\_adj\\_welding\\_cond\\_digital\(flag\\_reset=None, f\\_target=None, vel\\_target=None, wv\\_offset=None, wv\\_width\\_ratio=None, dynamic\\_cor=None, voltage\\_cor=None, job\\_number=None, synergic\\_id=None\)<sup>43</sup>](#)
- [app\\_weld\\_disable\\_digital\(\)\(p. 500\)](#)

<sup>41</sup><https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/5015203/>

app\_weld\_adj\_welding\_cond\_digital+flag\_reset+None+f\_target+None+vel\_target+None+wv\_offset+None+wv\_width\_ratio+None+dynamic\_cor+None+voltage\_cor+None+job\_number+None+synergic\_id+None

<sup>42</sup><https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/5015203/>

app\_weld\_adj\_welding\_cond\_digital+flag\_reset+None+f\_target+None+vel\_target+None+wv\_offset+None+wv\_width\_ratio+None+dynamic\_cor+None+voltage\_cor+None+job\_number+None+synergic\_id+None

<sup>43</sup><https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/5015203/>

app\_weld\_adj\_welding\_cond\_digital+flag\_reset+None+f\_target+None+vel\_target+None+wv\_offset+None+wv\_width\_ratio+None+dynamic\_cor+None+voltage\_cor+None+job\_number+None+synergic\_id+None

- [app\\_weld\\_weave\\_cond\\_trapezoidal\(wv\\_offset=\[0,0\], wv\\_ang=0, wv\\_param=\[0,1.5,0,-1.5,0.3,0.1,0.3,0.3,0.1,0.3\]\)\(p. 560\)](#)
- [app\\_weld\\_weave\\_cond\\_zigzag\(wv\\_offset=\[0,0\], wv\\_ang=0, wv\\_param=\[3,0.6\]\)\(p. 562\)](#)
- [app\\_weld\\_weave\\_cond\\_circular\(wv\\_offset=\[0,0\], wv\\_ang=0, wv\\_param=\[3,3,0.3,0.3\]\)\(p. 564\)](#)
- [app\\_weld\\_weave\\_cond\\_sinusoidal\(wv\\_offset=\[0,0\], wv\\_ang=0, wv\\_param=\[3,0.6\]\)\(p. 566\)](#)
- [moveL\(\)\(p. 62\)](#)
- [amoveL\(\)\(p. 103\)](#)
- [moveC\(\)\(p. 71\)](#)
- [amoveC\(\)\(p. 109\)](#)
- [moveB\(\)\(p. 85\)](#)
- [amoveB\(\)\(p. 119\)](#)
- [moveSX\(\)\(p. 81\)](#)
- [amoveSX\(\)\(p. 116\)](#)

### 12.3.13 app\_weld\_get\_welding\_cond\_digital()

#### 기능

통신방식 용접기를 이용한 용접 중 용접상태를 모니터링합니다. 모니터링 가능한 값은 현재 전압조정/동적인자조정/피딩속도/속도/위빙옵셋/에러번호/에러상태/와이어팁용착/옵션 및 측정전압/전류/피딩속도, 용접상태입니다. 기본 제공하는 모니터링 항목 외에 용접기로부터 모니터링 하고자 하는 신호를 추가하여 사용할 수 있습니다. 해당 신호는 인터페이스 `app_weld_set_interface_eip_m2r_other()`로 사전에 설정되어 있어야 됩니다. 추가적으로 용접상태(status) 인자를 통해 fail여부를 확인할 수 있습니다.

#### 리턴

값	설명
voltage_cor	현재 목표 전압조정(V) (조정값이 반영된 목표)
dynamic_cor	현재 목표 동적조정인자 (조정값이 반영된 목표)
f_target	현재 목표 피딩속도(m/min) (조정값이 반영된 목표)
vel_target	현재 목표 속도(mm/sec) (조정값이 반영된 목표) * 티치펜던트의 모니터링 출력단위와 다른 것에 유의할 것(Cm/min)
v_meas	현재 측정 전압(V)
c_meas	현재 측정 전류(A)
wv_offset[2]	현재 목표 옵셋(y, z방향, mm) (조정값이 반영된 목표)

값	설명
status	비용접:0, 용접(정상):1, 용접(이상):9, Dry-run:99
f_meas	현재 측정 피딩속도(mm/sec)
error_num	에러번호
wire_stick	와이어 티п 융착 상태(0: 정상, 1: 이상)
error	에러 상태(0: 정상, 1: 이상)
option1	옵션1 정보
option2	옵션2 정보
option3	옵션3 정보
option4	옵션4 정보
option5	옵션5 정보
option6	옵션6 정보
option7	옵션7 정보
option8	옵션8 정보
option9	옵션9 정보
option10	옵션10 정보
current_flow	전류흐름 상태(0: 정상, 1: 이상)
process_active	프로세스 활성화 상태(0: 정상, 1: 이상)
machine_ready	용접준비 상태(0: 정상, 1: 이상)

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 movej(posj(0,0,90,0,90,0),v=30,a=60)
2
3 pt1= posx(559, 434.5, 651.5, 45, 180, 45)
4 pt2= posx(559, 434.5, 151.5, 45, 180, 45)
5 pt3= posx(559, 0.0, 151.5, 45, 180, 45)
6
7 app_weld_enable_digital()
8
9 app_weld_set_weld_cond_digital(flag_dry_run=1, vel_target=16, vel_min=0.00,
   vel_max=16.67, welding_mode=3, s_2t=0, pulse_mode=0, wm_opt1=0,
   simulation=0, ts_opt1=0, ts_opt2=0, job_num=4, synergic_id=0,
   r_wire_feed_speed=15, voltage_correct=0, dynamic_correct=0, r_opt1=0,
   r_opt2=0, r_opt3=0, r_opt4=0, r_opt5=0, r_opt6=0, r_opt7=0, r_opt8=0,
   r_opt9=0, r_opt10=0, r_opt11=0, r_opt12=0, r_opt13=0, r_opt14=0, r_opt15=0)
10
11 movel(pt1, v=5, a=5, r=30, app_type=DR_MV_APP_WELD)
12 app_weld_adj_welding_cond_digital(flag_reset=0, f_target=10, vel_target=16,
   wv_offset=[20,30], wv_width_ratio=0.5,
   dynamic_cor=0, voltage_cor=0, job_number=5, synergic_id=4)
13 movel(pt2, v=5, a=5, r=30, app_type=DR_MV_APP_WELD)
14 app_weld_adj_welding_cond_digital(flag_reset=1)
15 amovel(pt3, v=5, a=5, app_type=DR_MV_APP_WELD)
16
17 [voltage_cor, dynamic_cor, f_target, vel_target, v_meas, c_meas,
18 wv_offset, status, f_meas, error_num, wire_stick,
19 error, opt1, opt2, opt3, opt4, opt5, opt6, opt7, opt8, opt9, opt10,
20 current_flow, process_active, machine_ready]=app_weld_get_welding_cond_digital();

```

```

21  while True:
22      if status == 9:
23          tp_popup("welding error!! ", DR_PM_ALARM, 1)
24          # 용접이상 발생 시 (status=9) 알람 발생
25      else :
26          if check_motion()==0:
27              break
28
29      app_weld_disable_digital()

```

## 관련 명령어

- [app\\_weld\\_set\\_interface\\_eip\\_r2m\\_process\(welding\\_start=\[0,0,0,0,0,0,0,0\], robot\\_ready=\[0,0,0,0,0,0,0,0\], error\\_reset=\[0,0,0,0,0,0,0,0\]\)](#)(p. 502)
- [app\\_weld\\_set\\_interface\\_eip\\_r2m\\_mode\(welding\\_mode=\[0,0,0,0,0,0,0,0\], s\\_2t=\[0,0,0,0,0,0,0,0\], pulse\\_mode=\[0,0,0,0,0,0,0,0\],wm\\_opt1=\[0,0,0,0,0,0,0,0\]\)](#)(p. 507)
- [app\\_weld\\_set\\_interface\\_eip\\_r2m\\_test\(gas\\_test=\[0,0,0,0,0,0,0,0\], inching\\_plus=\[0,0,0,0,0,0,0,0\], inching\\_minus=\[0,0,0,0,0,0,0,0\], blow\\_out\\_torch=\[0,0,0,0,0,0,0,0\], simulation=\[0,0,0,0,0,0,0,0\], ts\\_opt1=\[0,0,0,0,0,0,0,0\], ...\)](#)(p. 511)
- [app\\_weld\\_set\\_interface\\_eip\\_r2m\\_condition\(job\\_num=\[0,0,0,0,0,0,0,0\], synergic\\_id=\[0,0,0,0,0,0,0,0\], r\\_wire\\_feed\\_speed=\[0,0,0,0,0,0,0,0\], voltage\\_corret=\[0,0,0,0,0,0,0,0\], dynamic\\_correct=\[0,0,0,0,0,0,0,0\]\)](#)(p. 514)
- [app\\_weld\\_set\\_interface\\_eip\\_r2m\\_option\(opt1=\[0,0,0,0,0,0,0,0\], opt2=\[0,0,0,0,0,0,0,0\], opt3=\[0,0,0,0,0,0,0,0\], opt4=\[0,0,0,0,0,0,0,0\], opt5=\[0,0,0,0,0,0,0,0\], opt6=\[0,0,0,0,0,0,0,0\], ...\)](#)(p. 518)
- [app\\_weld\\_set\\_interface\\_eip\\_m2r\\_process\(current\\_flow=\[0,0,0,0,0,0,0,0\], process\\_active=\[0,0,0,0,0,0,0,0\], main\\_current=\[0,0,0,0,0,0,0,0\], machine\\_ready=\[0,0,0,0,0,0,0,0\], comm\\_ready=\[0,0,0,0,0,0,0,0\]\)](#)(p. 522)
- [app\\_weld\\_set\\_interface\\_eip\\_m2r\\_monitoring\(welding\\_voltage=\[0,0,0,0,0,0,0,0\], welding\\_current=\[0,0,0,0,0,0,0,0\], wire\\_feed\\_speed=\[0,0,0,0,0,0,0,0\], wire\\_stick=\[0,0,0,0,0,0,0,0\], error=\[0,0,0,0,0,0,0,0\] ...\)](#)(p. 525)
- [app\\_weld\\_set\\_interface\\_eip\\_m2r\\_other\(opt1=\[0,0,0,0,0,0,0,0\], opt2=\[0,0,0,0,0,0,0,0\], opt3=\[0,0,0,0,0,0,0,0\], opt4=\[0,0,0,0,0,0,0,0\], opt5=\[0,0,0,0,0,0,0,0\], opt6=\[0,0,0,0,0,0,0,0\], opt7=\[0,0,0,0,0,0,0,0\], ...\)](#)(p. 529)
- [app\\_weld\\_enable\\_digital\(\)](#)(p. 498)
- [app\\_weld\\_get\\_welding\\_cond\\_digital\(\)](#)<sup>44</sup>
- [app\\_weld\\_get\\_welding\\_cond\\_digital\(\)](#)<sup>45</sup>
- [app\\_weld\\_get\\_welding\\_cond\\_digital\(\)](#)<sup>46</sup>
- [app\\_weld\\_disable\\_digital\(\)](#)(p. 500)

<sup>44</sup> <https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/5015207/>

app\_weld\_get\_welding\_cond\_digital

<sup>45</sup> <https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/5015207/>

app\_weld\_get\_welding\_cond\_digital

<sup>46</sup> <https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/5015207/>

app\_weld\_get\_welding\_cond\_digital

- [app\\_weld\\_weave\\_cond\\_trapezoidal\(wv\\_offset=\[0,0\], wv\\_ang=0, wv\\_param=\[0,1.5,0,-1.5,0.3,0.1,0.3,0.3,0.1,0.3\]\)\(p. 560\)](#)
- [app\\_weld\\_weave\\_cond\\_zigzag\(wv\\_offset=\[0,0\], wv\\_ang=0, wv\\_param=\[3,0.6\]\)\(p. 562\)](#)
- [app\\_weld\\_weave\\_cond\\_circular\(wv\\_offset=\[0,0\], wv\\_ang=0, wv\\_param=\[3,3,0.3,0.3\]\)\(p. 564\)](#)
- [app\\_weld\\_weave\\_cond\\_sinusoidal\(wv\\_offset=\[0,0\], wv\\_ang=0, wv\\_param=\[3,0.6\]\)\(p. 566\)](#)
- [moveL\(\)\(p. 62\)](#)
- [amoveL\(\)\(p. 103\)](#)
- [moveC\(\)\(p. 71\)](#)
- [amoveC\(\)\(p. 109\)](#)
- [moveB\(\)\(p. 85\)](#)
- [amoveB\(\)\(p. 119\)](#)
- [moveSX\(\)\(p. 81\)](#)
- [amoveSX\(\)\(p. 116\)](#)

### **12.3.14 app\_weld\_enable\_analog(ch\_v\_out=[1,0], spec\_v\_out=[0,0,0,0], ch\_f\_out=[2,0], spec\_f\_out=[0,0,0,0], ch\_v\_in=[1,0], spec\_v\_in=[0,0,0,0], ch\_c\_in=[2,0], spec\_c\_in=[0,0,0,0], ch\_arc\_on=1, ch\_gas\_on=2, ch\_inching\_fwd=3, ch\_inching\_bwd=4, ...)**

#### **기능**

아날로그 용접기능을 활성화합니다. 아날로그 입출력 및 디지털 신호출력 방식으로 연결 가능한 용접기의 연결 및 환경정보를 입력 인자로 입력합니다.

대상 용접기는 아날로그 인터페이스 방식을 지원하여 연결된 제어기의 아날로그 출력 채널로부터 목표전류 및 목표 전압지령을 입력 받을 수 있어야 합니다. 물리적으로 연결된 아날로그 채널의 채널번호(1 또는 2)와 출력모드(전류/전압)를 ch\_v\_out, ch\_f\_out에 설정하세요. 제어기의 아날로그 입/출력 범위는 전압모드 설정 시 0~10V, 전류모드 설정 시 4~20mA입니다. 각 채널별 설정모드 및 출력범위가 용접기의 입력사양 및 범위와 호환되도록 하세요. (예를 들어 용접기의 목표값 입력 범위가 0~10V라고 하면, 제어기의 출력채널을 전압모드(0~10V 출력범위)로 설정하는 것이 적절합니다. 또 다른 예로 용접기의 입력채널 사양이 2~15V라고 하면, 대응되는 제어기의 아날로그 채널은 전류모드 (4~20mA 출력범위)로 설정한 후 출력라인에 75옴의 저항을 연결하여 3~15V 범위의 전압을 출력할 수 있도록 연결합니다. 이 경우 제어기로 설정할 수 없는 2V~3V사이의 범위는 지령을 줄 수 없게됩니다.) 가능한 한 용접기에서 요구되는 입력범위를 많이 포함할 수 있도록 설정하는 것이 좋습니다.

제어기의 아날로그 출력 최대 및 최소 범위와 용접기의 출력 최대 및 최소 범위를 spec\_v\_out, spec\_f\_out에 설정 합니다.

spec\_v\_out/spec\_f\_out의 첫째 항목 = WO\_min  
 spec\_v\_out/spec\_f\_out의 둘째 항목 = CO\_min  
 spec\_v\_out/spec\_f\_out의 셋째 항목 = WO\_max  
 spec\_v\_out/spec\_f\_out의 넷째 항목 = CO\_max

여기서, WO\_min, WO\_max는 용접기의 최소, 최대 출력사양이며, CO\_min, CO\_max는 각각 WO\_min, WO\_max에 대응되는 제어기의 아날로그 출력 값입니다.

### **i 알아두기**

용접기에서 출력되는 용접전류는 와이어 피딩속도는 물론 모재의 재질, 용접와이어의 재질/종류/토출길이, 용접전압등에 의해 변동되며 이는 용접기 또는 별도로 장착한 전류센서를 연결하여 확인하여야 합니다.

용접중인 전압/전류측정값을 확인하기 위해 아날로그 출력방식의 용접기 또는 별도의 센서를 연결하여야 합니다. 이에 대응하는 제어기의 아날로그 입력 채널번호 및 입력모드를 ch\_v\_in, ch\_c\_in에 설정합니다..

제어기의 아날로그 입력 최대 및 최소 범위와 센서의 측정 최대 및 최소 범위를 spec\_v\_in,, spec\_c\_in에 설정합니다.

spec\_v\_in/spec\_c\_in의 첫째 항목 = SO\_min

spec\_v\_in/spec\_c\_in의 둘째 항목 = CI\_min

spec\_v\_in/spec\_c\_in의 셋째 항목 = SO\_max

spec\_v\_in/spec\_c\_in의 네째 항목 = CI\_max

여기서, SO\_min, SO\_max는 각각 센서의 최소, 최대 측정 값이며, CI\_min, CI\_max는 각각 SO\_min, SO\_max에 대응되는 제어기의 입력값입니다.

디지털 접점방식으로 용접기와 연결되는 ARC-ON/OFF(용접출력신호-시작/종료), GAS-ON/OFF(가스출력신호-시작/종료), INCHING-Forward-ON/OFF(정방향와이어송급신호-시작/종료), INCHING-Backward-ON/OFF(역방향와이어송급신호-시작/종료), BlowOut-ON/OFF(토치청소가스출력신호-시작/종료) 채널번호를 설정하십시오. ARC-ON/OFF 신호 외의 신호출력은 용접기의 해당기능 지원여부에 따라서 선택적으로 입력하십시오.

## 인수

인수명	자료형	기본값	설명
ch_v_out	list(int[2])	1	전압지령 아날로그 출력 채널 (1~2) 지정하지 않는 경우 : 0
		0	0:전류모드(4~20mA) 1:전압모드(0~10V)
spec_v_out	list(float[4])	0	용접기출력전압(V) 최소값(a)
		0	(a)에 대응하는 제어기출력값
		0	용접기출력전압(V) 최대값(b)

인수명	자료형	기본값	설명
		0	(b)에 대응하는 제어기출력값
ch_f_out	list(int[2])	2	피딩속도지령 아날로그 출력 채널 (1~2) 지정하지 않는 경우 : 0
		0	0:전류모드(4~20mA) 1:전압모드(0~10V)
spec_f_out	list(float[4])	0	피딩속도(m/min) 최소값(c)
		0	(c)에 대응하는 제어기출력값
		0	피딩속도(m/min) 최대값(d)
		0	(d)에 대응하는 제어기출력값
ch_v_in	list(int[2])	1	전압센서 아날로그 입력 채널 (1~2) 센서가 없는 경우 : 0
		0	0:전류모드(4~20mA) 1:전압모드(0~10V)
spec_v_in	list(float[4])	0	전압센서입력(V) 최소값(e)
		0	(e)에 대응하는 제어기입력값
		0	전압센서입력(V) 최대값(f)
		0	(f)에 대응하는 제어기입력값
ch_c_in	list(int[2])	2	전류센서 아날로그 입력 채널 (1~2) 센서가 없는 경우 : 0
		0	0:전류모드(4~20mA) 1:전압모드(0~10V)

인수명	자료형	기본값	설명
spec_c_in	list(float[4])	0	전류센서입력(A) 최소값(g)
		0	(g)에 대응하는 제어기입력값
		0	전류센서입력(A) 최대값(h)
		0	(h)에 대응하는 제어기입력값
ch_arc_on	int	1	용접출력 디지털 출력 채널(1~16)
ch_gas_on	int	2	보호가스출력 디지털 출력 채널 (1~16) 연결하지 않는 경우 : 0
ch_inching_fwd	int	3	용접와이어 정방향 토출 디지털 출력 채널(1~16) 연결하지 않는 경우 : 0
ch_inching_bwd	int	4	용접와이어 역방향 토출 디지털 출력 채널(1~16) 연결하지 않는 경우 : 0
ch_blow_out	int	5	토치청소가스출력 디지털 출력 채널(1~16) 연결하지 않는 경우 : 0

## 리턴

값	설명
0	용접기능 활성화 성공
음의값	용접기능 활성화 실패

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 app_weld_enable_analog(ch_v_out=[1,1], spec_v_out=[0,0,300,10], ch_f_out =[2
2 , spec_f_out =[0,0,40,10], ch_v_in =[1,1], spec_v_in =[0,0,300,10], ch_c_in
3 =[2,1],
4 spec_c_in=[0,0,40,10], ch_arc_on=1, ch_gas_on=2, ch_inching_fwd=3,
5 ch_inching_bwd=4, ch_blow_out=5)
6 # 전압출력(1번채널, 전압모드), 용접기전압사양(최소/최대)=(0~300)
7 # 피딩속도출력(2번채널, 전압모드), 피딩속도사양(최소/최대)=(0~40)
8 # 전압센싱(1번채널, 전압모드), 센서측정사양(최소/최대)=(0~300)
9 # 전류센싱(2번채널, 전압모드), 센서사양(최소/최대)=(0~40)
10 # 용접시작신호(1번채널), 가스 출력신호(2번채널), 와이어정방향토출신호(3번채널),
11 # 와이어역방향토출신호(4번채널), 토치청소가스출력신호(5번채널)
app_weld_disable_analog()

```

## 관련 명령어

- [app\\_weld\\_disable\\_analog\(\)](#)(p. 550)
- [app\\_weld\\_set\\_weld\\_cond\\_analog\(flag\\_dry\\_run=0, v\\_target=0, f\\_target=0, vel\\_target=0, vel\\_min=0, vel\\_max=0, weld\\_proc\\_param=\[0.2,0.2,0.5,0.5,0.5,0.2,0.2,0.5,0.5\]\)](#)(p. 551)
- [app\\_weld\\_weave\\_cond\\_trapezoidal\(wv\\_offset=\[0,0\], wv\\_ang=0, wv\\_param=\[0,1.5,0,-1.5,0.3,0.1,0.3,0.3,0.1,0.3\]\)](#)(p. 560)
- [app\\_weld\\_weave\\_cond\\_zigzag\(wv\\_offset=\[0,0\], wv\\_ang=0, wv\\_param=\[3,0.6\]\)](#)(p. 562)
- [app\\_weld\\_weave\\_cond\\_circular\(wv\\_offset=\[0,0\], wv\\_ang=0, wv\\_param=\[3,3,0.3,0.3\]\)](#)(p. 564)
- [app\\_weld\\_weave\\_cond\\_sinusoidal\(wv\\_offset=\[0,0\], wv\\_ang=0, wv\\_param=\[3,0.6\]\)](#)(p. 566)

- app\_weld\_adj\_welding\_cond\_analog(flag\_reset=0, v\_target=None, f\_target=None, vel\_target=None, wv\_offset=None, wv\_width\_ratio=None)(p. 555)
- app\_weld\_get\_welding\_cond\_analog()(p. 557)

### 12.3.15 app\_weld\_disable\_analog()

#### 기능

아날로그 용접기능을 비활성화합니다..

#### 리턴

값	설명
0	성공
음수	실패

#### 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

#### 예제

```

1   app_weld_enable_analog(ch_v_out=[1,1], spec_v_out=[0,0,300,10], ch_f_out =[2
2   ,
3   spec_f_out =[0,0,40,10], ch_v_in =[1,1], spec_v_in =[0,0,300,10], ch_c_in
= [2,1],
3   spec_c_in=[0,0,40,10], ch_arc_on=1, ch_gas_on=2, ch_inching_fwd=3,
ch_inching_bwd=4, ch_blow_out=5)

```

4	
5	app_weld_disable_analog()

## 관련 명령어

- app\_weld\_enable\_analog(ch\_v\_out=[1,0], spec\_v\_out=[0,0,0,0], ch\_f\_out=[2,0], spec\_f\_out=[0,0,0,0], ch\_v\_in=[1,0], spec\_v\_in=[0,0,0,0], ch\_c\_in=[2,0], spec\_c\_in=[0,0,0,0], ch\_arc\_on=1, ch\_gas\_on=2, ch\_inching\_fwd=3, ch\_inching\_bwd=4, ...)(p. 545)...
- app\_weld\_set\_weld\_cond\_analog(flag\_dry\_run=0, v\_target=0, f\_target=0, vel\_target=0, vel\_min=0, vel\_max=0, weld\_proc\_param=[0.2,0.2,0.5,0.5,0.5,0.2,0.2,0.5,0.5])(p. 551)
- app\_weld\_weave\_cond\_trapezoidal(wv\_offset=[0,0], wv\_ang=0, wv\_param=[0,1.5,0,-1.5,0.3,0.1,0.3,0.3,0.1,0.3])(p. 560)
- app\_weld\_weave\_cond\_zigzag(wv\_offset=[0,0], wv\_ang=0, wv\_param=[3,0.6])(p. 562)
- app\_weld\_weave\_cond\_circular(wv\_offset=[0,0], wv\_ang=0, wv\_param=[3,3,0.3,0.3])(p. 564)
- app\_weld\_weave\_cond\_sinusoidal(wv\_offset=[0,0], wv\_ang=0, wv\_param=[3,0.6])(p. 566)
- app\_weld\_adj\_welding\_cond\_analog(flag\_reset=0, v\_target=None, f\_target=None, vel\_target=None, wv\_offset=None, wv\_width\_ratio=None)(p. 555)
- app\_weld\_get\_welding\_cond\_analog()(p. 557)

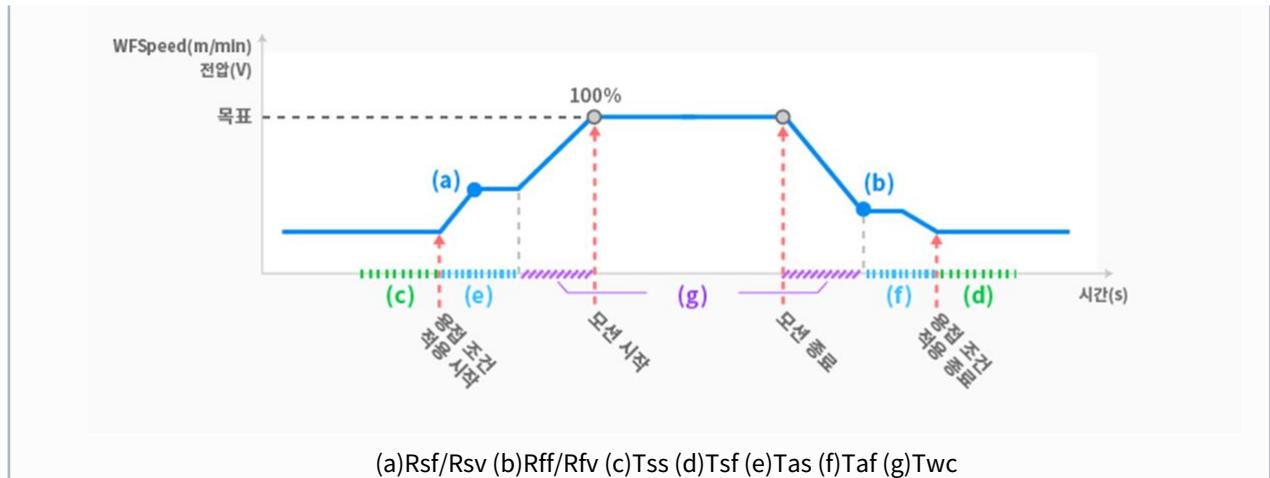
### 12.3.16 app\_weld\_set\_weld\_cond\_analog(flag\_dry\_run=0, v\_target=0, f\_target=0, vel\_target=0, vel\_min=0, vel\_max=0, weld\_proc\_param=[0.2,0.2,0.5,0.5,0.5,0.2,0.2,0.5,0.5])

## 기능

아날로그 용접조건을 설정합니다. 용접조건은 용접기능 활성화 (app\_weld\_enable\_analog()) 부터 비활성화 (app\_weld\_disable\_analog())로 정의한 용접구간 내에서만 유효하며 이외에 실행 시 에러가 발생합니다. 용접조건 중 용접공정변수(weld\_proc\_param)는 용접의 시작/종료 시의 가스/조건유지시간 등의 상세한 조건을 나타냅니다. 아래의 그림을 참조하여 입력하세요. 하나의 용접구간 내에서는 하나의 용접조건만 허용되며 용접 중 app\_weld\_adj\_welding\_cond\_analog() 명령어를 통해 용접조건을 조정하거나, 티치펜던트의 용접조건 조정 popup에서 전압/피딩속도/속도 (및 위빙옵셋)를 조정할 수 있습니다. 다만 티치펜던트에서의 용접조건 조정은 명령어를 통한 용접조건 조정상태가 RESET상태 (즉, app\_weld\_set\_weld\_cond\_analog()으로 지정한 용접조건 설정) 인 경우에만 가능합니다.

## ● 알아두기

용접기에서 출력되는 용접전류는 와이어 피딩속도는 물론 모재의 재질, 용접와이어의 재질/종류/토출길이, 용접전압등에 의해 변동되며 이는 용접기 또는 별도로 장착한 전류센서를 연결하여 확인하여야 합니다.



## 인수

인수명	자료형	기본값	설명
flag_dry_run	int	0	모의용접(Dry-run) 모드 실용접(0) 모의용접(1) : 모션/위빙/옵셋만 진행
v_target	float	0	목표전압 (V)
f_target	float	0	목표피딩속도 (m/min)
vel_target	float	0	목표속도 (mm/sec) * 티치펜던트의 입력단위와 다른 것에 유의할 것(Cm/min)
vel_min	float	0	목표속도 조정 최소값 (mm/sec) * 티치펜던트의 입력단위와 다른 것에 유의할 것(Cm/min)
vel_max	float	0	목표속도 조정 최대값 (mm/sec) * 티치펜던트의 입력단위와 다른 것에 유의할 것(Cm/min)
weld_proc_param	list(float[9])	0.2	Rsf (피딩속도 시작조건/목표조건 비) (0 < Rsf <= 1)
		0.2	Rsv (전압 시작조건/목표조건 비) (0 < Rsv <= 1)

인수명	자료형	기본값	설명
		0.5	Tss (용접전 보호가스 토출시간, sec) (0<= Tss)
		0.5	Tas (용접시작조건 유지시간, sec) (0<= Twc)
		0.5	Twc (용접본조건 변동시간, sec) (0<= Twc)
		0.2	Rff (피딩속도 끝조건/목표조건 비) (0< Rff <= 1)
		0.2	Rfv (전압 끝조건/목표조건 비) (0< Rfv <= 1)
		0.5	Taf (용접끝조건 유지시간, sec) (0<= Taf)
		0.5	Tsf (용접후 보호가스 토출시간, sec) (0<= Tsf)

## 리턴

값	설명
0	설정 성공
음수값	설정 실패

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시

예외	설명
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 app_weld_enable_analog(ch_v_out=[1,1], spec_v_out=[0,0,300,10], ch_f_out =[2
2 ,
3 spec_f_out =[0,0,40,10], ch_v_in =[1,1], spec_v_in =[0,0,300,10], ch_c_in
4 =[2,1],
5 spec_c_in=[0,0,40,10], ch_arc_on=1, ch_gas_on=2, ch_inching_fwd=3,
6 ch_inching_bwd=4, ch_blow_out=5)
7 # 목표전압/피딩속도 = 24V, 20m/min, 용접속도=60mm/sec(=1Cm/min), 실용접, 용접공정변
8 수 기본값 사용
  app_weld_disable_analog()

```

## 관련 명령어

- [app\\_weld\\_set\\_weld\\_cond\\_analog\(flag\\_dry\\_run=0, v\\_target=0, f\\_target=0, vel\\_target=0, vel\\_min=0, vel\\_max=0, weld\\_proc\\_param=\[0.2,0.2,0.5,0.5,0.5,0.2,0.2,0.5,0.5\]\)<sup>47</sup>](#)
- [app\\_weld\\_set\\_weld\\_cond\\_analog\(flag\\_dry\\_run=0, v\\_target=0, f\\_target=0, vel\\_target=0, vel\\_min=0, vel\\_max=0, weld\\_proc\\_param=\[0.2,0.2,0.5,0.5,0.5,0.2,0.2,0.5,0.5\]\)<sup>48</sup>](#)
- [app\\_weld\\_weave\\_cond\\_trapezoidal\(wv\\_offset=\[0,0\], wv\\_ang=0, wv\\_param=\[0,1.5,0,-1.5,0.3,0.1,0.3,0.3,0.1,0.3\]\)\(p. 560\)](#)
- [app\\_weld\\_weave\\_cond\\_zigzag\(wv\\_offset=\[0,0\], wv\\_ang=0, wv\\_param=\[3,0.6\]\)\(p. 562\)](#)
- [app\\_weld\\_weave\\_cond\\_circular\(wv\\_offset=\[0,0\], wv\\_ang=0, wv\\_param=\[3,3,0.3,0.3\]\)\(p. 564\)](#)
- [app\\_weld\\_weave\\_cond\\_sinusoidal\(wv\\_offset=\[0,0\], wv\\_ang=0, wv\\_param=\[3,0.6\]\)\(p. 566\)](#)
- [app\\_weld\\_adj\\_welding\\_cond\\_analog\(flag\\_reset=0, v\\_target=None, f\\_target=None, vel\\_target=None, wv\\_offset=None, wv\\_width\\_ratio=None\)\(p. 555\)](#)

<sup>47</sup><https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/5015219/>

app\_weld\_set\_weld\_cond\_analog+flag\_dry\_run+0+v\_target+0+f\_target+0+vel\_target+0+vel\_min+0+vel\_max+0+weld\_proc\_param+0.2+0.2+0.5+0.5+0.2+0.2+0.5+0.5

<sup>48</sup><https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/5015219/>

app\_weld\_set\_weld\_cond\_analog+flag\_dry\_run+0+v\_target+0+f\_target+0+vel\_target+0+vel\_min+0+vel\_max+0+weld\_proc\_param+0.2+0.2+0.5+0.5+0.2+0.2+0.5+0.5

- app\_weld\_get\_welding\_cond\_analog()(p. 557)

### 12.3.17 app\_weld\_adj\_welding\_cond\_analog(flag\_reset=0, v\_target=None, f\_target=None, vel\_target=None, wv\_offset=None, wv\_width\_ratio=None)

#### 기능

아날로그 용접 중 용접 및 위빙조건을 조정합니다. 일반적으로 연속된 경로에서 구간별로 용접조건을 변경하고자 할 때 모션명령어(movel(), movec(), moveb(), movesx()) 호출 직전에 사용합니다. 본 명령어로 조정인자를 입력한 경우 해당하는 용접 및 위빙조건이 조정되며 이 때에는 TP의 용접모니터링 정보창에서 용접/위빙조건을 실시간으로 조정할 수 없습니다. 조정조건에서 본조건(app\_weld\_set\_weld\_cond\_analog() 및 app\_weld\_weave\_cond\_trapezoidal() 등으로 설정한 용접/위빙조건)으로 복귀하려면 flag\_reset=1로 실행하세요. flag\_reset=1 설정 시 TP에서 실시간으로 조정한 최종 조건으로 복귀되며(실시간으로 조정불가능한 위빙폭의 비(wv\_width\_ratio)는 1로 변경됩니다.) TP에서 용접조건을 실시간으로 조정할 수 있습니다.

#### 인수

인수명	자료형	기본값	설명
flag_reset	int	0	0 : 조정값 적용 1 : 기준목표(app_weld_set_weld_cond_analog()) 값 적용
v_target	float	-	목표전압 (V)
f_target	float	-	피딩속도 (m/min)
vel_target	float	-	목표속도 (mm/sec) * 티치펜던트의 입력단위와 다른 것에 유의할 것(Cm/min)
wv_offset	float[2]	-	위빙좌표계-y방향 옵셋 (mm)
		-	위빙좌표계-z방향 옵셋 (mm)
wv_width_ratio	float	-	변경위빙폭/설정위빙폭 의 비 (0~2)

#### i 알아두기

인수 v\_target/f\_target/vel\_target/wv\_offset/wv\_width\_ratio 중 값을 지정하지 않는 조건은 현재의 조건(실시간으로 조정한 조건 포함)이 유지되므로 조정을 원하는 인수만 설정하세요. 단 wv\_offset의 경우 y방향 또는 z방향 중 하나만 조정하더라도 배열의 두 값을 모두 입력하여야 합니다.

## 리턴

값	설명
0	설정 성공
음수값	설정 실패

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 movej(posj(0,0,90,0,90,0),v=30,a=60)
2
3 pt1= posx(559, 434.5, 651.5, 45, 180, 45)
4 pt2= posx(559, 434.5, 151.5, 45, 180, 45)
5 pt3= posx(559, 0.0, 151.5, 45, 180, 45)
6
7 app_weld_enable_analog(ch_v_out=[1,1], spec_v_out=[0,0,300,10], ch_f_out =[2
8 ,
9 spec_f_out =[0,0,40,10], ch_v_in =[1,1], spec_v_in =[0,0,300,10], ch_c_in
10 =[2,1],
11 spec_c_in=[0,0,40,10], ch_arc_on=1, ch_gas_on=2, ch_inching_fwd=3,
12 ch_inching_bwd=4, ch_blow_out=5)
13
14 app_weld_set_weld_cond_analog(flag_dry_run=1, v_target=24, f_target=20,
15 vel_target=60, vel_min=10,
vel_max=100, weld_proc_param=[0.2,0.2,0.5,0.5,0.5,0.2,0.2,0.5,0.5])
```

```

16 app_weld_adj_welding_cond_analog(flag_reset=0, v_target=20, f_target=10,
17 vel_target=30, wv_offset=[20,10], wv_width_ratio=0.5)
18 movel(pt2, v=5, a=5, r=30, app_type=DR_MV_APP_WELD)
19 app_weld_adj_welding_cond_analog(flag_reset=1)
20 movel(pt3, v=5, a=5, app_type=DR_MV_APP_WELD)
# 시작위치 → pt1 : 초기설정용접조건 적용 (24V, 20m/min)
21 # pt1 → pt2 : 조정조건 적용 (20V, 10m/min)
22 # pt2 → pt3 : 초기설정 적용 (24V, 20m/min)
23
24 app_weld_disable_analog()

```

## 관련 명령어

- `app_weld_enable_analog(ch_v_out=[1,0], spec_v_out=[0,0,0,0], ch_f_out=[2,0], spec_f_out=[0,0,0,0], ch_v_in=[1,0], spec_v_in=[0,0,0,0], ch_c_in=[2,0], spec_c_in=[0,0,0,0], ch_arc_on=1, ch_gas_on=2, ch_inching_fwd=3, ch_inching_bwd=4, ...)`(p. 545)
- `app_weld_set_weld_cond_analog(flag_dry_run=0, v_target=0, f_target=0, vel_target=0, vel_min=0, vel_max=0, weld_proc_param=[0.2,0.2,0.5,0.5,0.5,0.2,0.2,0.5,0.5])`(p. 551)
- `app_weld_weave_cond_trapezoidal(wv_offset=[0,0], wv_ang=0, wv_param=[0,1.5,0,-1.5,0.3,0.1,0.3,0.3,0.1,0.3])`(p. 560)
- `app_weld_weave_cond_zigzag(wv_offset=[0,0], wv_ang=0, wv_param=[3,0.6])`(p. 562)
- `app_weld_weave_cond_circular(wv_offset=[0,0], wv_ang=0, wv_param=[3,3,0.3,0.3])`(p. 564)
- `app_weld_weave_cond_sinusoidal(wv_offset=[0,0], wv_ang=0, wv_param=[3,0.6])`(p. 566)
- `app_weld_adj_welding_cond_analog(flag_reset=0, v_target=None, f_target=None, vel_target=None, wv_offset=None, wv_width_ratio=None)`<sup>49</sup>
- `app_weld_get_welding_cond_analog()`(p. 557)
- `movel()`(p. 62)
- `amovel()`(p. 103)
- `movec()`(p. 71)
- `amovec()`(p. 109)
- `moveb()`(p. 85)
- `amoveb()`(p. 119)
- `movesx()`(p. 81)
- `amovesx()`(p. 116)

### 12.3.18 app\_weld\_get\_welding\_cond\_analog()

#### 기능

아날로그 용접 중 용접상태를 모니터링합니다. 모니터링 가능한 값은 현재 목표전압/전류/속도/위빙옵셋/Digital 출력 신호 및 측정전압/전류, 용접상태입니다. 측정전압/전류 설정이 되어 있지 않은 경우 (app\_weld\_enable()) 시

<sup>49</sup><https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/5015223/>

app\_weld\_adj\_welding\_cond\_analog+flag\_reset+0+v\_target+None+f\_target+None+vel\_target+None+wv\_offset+None+wv\_width\_ratio+None

ch\_v\_in, ch\_c\_in 이 미설정된 경우) 해당 값은 현재 목표전압(v\_target)/전류(c\_target) 값과 동일하게 출력됩니다.  
추가적으로 용접상태(status) 인자를 통해 fail여부를 확인할 수 있습니다.

## 리턴

값	설명
v_target	현재 목표 전압(V) (조정값이 반영된 목표)
c_target	현재 목표 전류(A) (조정값이 반영된 목표)
f_target	현재 목표 피딩속도(m/min) (조정값이 반영된 목표)
vel_target	현재 목표 속도(mm/sec) (조정값이 반영된 목표) * 티치펜던트의 모니터링 출력단위와 다른 것에 유의할 것(cm/min)
v_meas	현재 측정 전압(V)
c_meas	현재 측정 전류(A)
wv_offset[2]	현재 목표 움셋(y, z방향, mm) (조정값이 반영된 목표)
sig_out[4]	Digital 출력신호 (arc_on, gas_on, inching_fwd, inching_bwd)
status	비용접:0, 용접(정상):1, 용접(이상):9, Dry-run:99

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 movej(posj(0,0,90,0,90,0),v=30,a=60)
2
3 pt1= posx(559, 434.5, 651.5, 45, 180, 45)
4 pt2= posx(559, 434.5, 151.5, 45, 180, 45)
5 pt3= posx(559, 0.0, 151.5, 45, 180, 45)
6
7 app_weld_enable_analog(ch_v_out=[1,1], spec_v_out=[0,0,300,10], ch_f_out =[2
8 , spec_f_out =[0,0,40,10], ch_v_in =[1,1], spec_v_in =[0,0,300,10], ch_c_in
9 =[2,1],
10 spec_c_in=[0,0,40,10], ch_arc_on=1, ch_gas_on=2, ch_inching_fwd=3,
11 ch_inching_bwd=4, ch_blow_out=5)
12 app_weld_set_weld_cond_analog(flag_dry_run=1, v_target=24, f_target=20,
13 vel_target=60, vel_min=10,
14 vel_max=100, weld_proc_param=[0.2,0.2,0.5,0.5,0.5,0.2,0.2,0.5,0.5])
15 movel(pt1, v=5, a=5, r=30, app_type=DR_MV_APP_WELD)
16 app_weld_adj_welding_cond_analog(flag_reset=0, v_target=20, f_target=10,
17 vel_target=30, wv_offset=[20,10], wv_width_ratio=0.5)
18 movel(pt2, v=5, a=5, r=30, app_type=DR_MV_APP_WELD)
19 app_weld_adj_welding_cond_analog(flag_reset=1)
20 amovel(pt3, v=5, a=5, app_type=DR_MV_APP_WELD)
21
22 while True:
23     Vt, Ct, Ft, velt,Vm, Cm, Off, Dout, status =
24         app_weld_get_welding_cond_analog()
25     if status == 9:
26         tp_popup("welding error!! ", DR_PM_ALARM, 1)
27         # 용접이상 발생 시 (status=9) 알람 발생
28     else :
29         if check_motion()==0:
30             break
31
32 app_weld_disable_analog()

```

## 관련 명령어

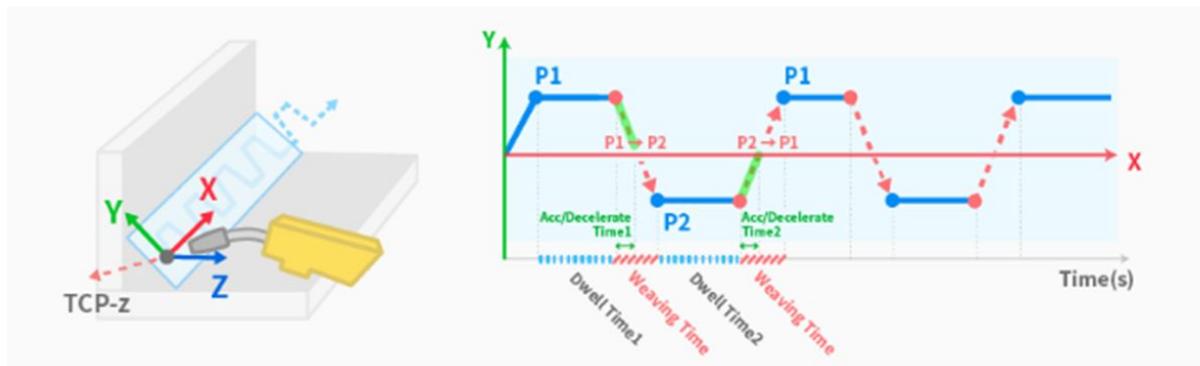
- [app\\_weld\\_enable\\_analog\(ch\\_v\\_out=\[1,0\], spec\\_v\\_out=\[0,0,0,0\], ch\\_f\\_out=\[2,0\], spec\\_f\\_out=\[0,0,0,0\], ch\\_v\\_in=\[1,0\], spec\\_v\\_in=\[0,0,0,0\], ch\\_c\\_in=\[2,0\], spec\\_c\\_in=\[0,0,0,0\], ch\\_arc\\_on=1, ch\\_gas\\_on=2, ch\\_inching\\_fwd=3, ch\\_inching\\_bwd=4, ...\)](#)(p. 545)
- [app\\_weld\\_set\\_weld\\_cond\\_analog\(flag\\_dry\\_run=0, v\\_target=0, f\\_target=0, vel\\_target=0, vel\\_min=0, vel\\_max=0, weld\\_proc\\_param=\[0.2,0.2,0.5,0.5,0.5,0.2,0.2,0.5,0.5\]\)](#)(p. 551)
- [app\\_weld\\_weave\\_cond\\_trapezoidal\(wv\\_offset=\[0,0\], wv\\_ang=0, wv\\_param=\[0,1.5,0,-1.5,0.3,0.1,0.3,0.3,0.1,0.3\]\)](#)(p. 560)
- [app\\_weld\\_weave\\_cond\\_zigzag\(wv\\_offset=\[0,0\], wv\\_ang=0, wv\\_param=\[3,0.6\]\)](#)(p. 562)

- `app_weld_weave_cond_circular(wv_offset=[0,0], wv_ang=0, wv_param=[3,3,0.3,0.3])(p. 564)`
- `app_weld_weave_cond_sinusoidal(wv_offset=[0,0], wv_ang=0, wv_param=[3,0.6])(p. 566)`
- `app_weld_get_welding_cond_analog()`<sup>50</sup>
- `app_weld_get_welding_cond_analog()`<sup>51</sup>

### 12.3.19 `app_weld_weave_cond_trapezoidal(wv_offset=[0,0], wv_ang=0, wv_param=[0,1.5,0,-1.5,0.3,0.1,0.3,0.3,0.1,0.3])`

#### 기능

사다리꼴 위빙조건을 설정합니다. 위빙조건은 용접기능 활성화 (`app_weld_enable_analog()` / `app_weld_enable_digital()`)부터 비활성화(`app_weld_disable_analog()` / `app_weld_disable_digital()`)로 정의한 용접구간 내에서만 유효하며 이외에 실행 시 에러가 발생합니다. 위빙조건은 용접경로의 진행방향을 위빙x축으로, 위빙x축에서 TCP-z방향을 벡터곱(cross-product)한 방향을 위빙y축으로 정의하는 위빙좌표계에서 정의됩니다. 좌표계 및 위빙 설정 인자는 아래 그림을 참조하세요. 하나의 용접구간 내에서는 하나의 위빙조건만 허용되며 용접 중 `app_weld_adj_welding_cond_analog()` / `app_weld_set_weld_cond_digital()` 명령어를 통해 옵셋 또는 위빙폭을 조정하거나, 티치펜던트의 용접조건 조정 popup에서 (전압/전류/속도 및) 옵셋을 조정할 수 있습니다. 다만 티치펜던트에서의 용접조건 조정은 명령어를 통한 용접조건 조정상태가 RESET상태(즉, `app_weld_set_weld_cond_analog()` / `app_weld_set_weld_cond_digital()`으로 지정한 용접조건 설정)인 경우에만 가능합니다.



#### 인수

인수명	자료형	기본값	설명
<code>wv_offset</code>	<code>float[2]</code>	0	위빙좌표계-y방향 옵셋 (mm)
		0	위빙좌표계-z방향 옵셋 (mm)

<sup>50</sup> <https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/5015227/>

`app_weld_get_welding_cond_analog`

<sup>51</sup> <https://doosanrobotics-manual.atlassian.net/wiki/spaces/ProgrammingMT/pages/5015227/>

`app_weld_get_welding_cond_analog`

인수명	자료형	기본값	설명
wv_angle	float	0	위빙좌표계-x축기준 위빙평면 회전각 (deg)
wv_param	list(float[10])	0	위빙점1-x (mm)
		1.5	위빙점1-y (mm)
		0	위빙점2-x (mm)
		-1.5	위빙점2-y (mm)
		0.3	위빙점1→2시간(sec)
		0.1	위빙점1→2가감속시간(sec)
		0.3	위빙점1 드웰시간(sec)
		0.3	위빙점2→1시간(sec)
		0.1	위빙점2→1가감속시간(sec)
		0.3	위빙점2 드웰시간(sec)

## 리턴

값	설명
0	설정 성공
음수값	설정 실패

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시

예외	설명
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 app_weld_enable_analog(ch_v_out=[1,1], spec_v_out=[0,0,300,10], ch_f_out =[2
2 ,
3 spec_f_out =[0,0,40,10], ch_v_in =[1,1], spec_v_in =[0,0,300,10], ch_c_in
4 = [2,1],
5 spec_c_in=[0,0,40,10], ch_arc_on=1, ch_gas_on=2, ch_inching_fwd=3,
6 ch_inching_bwd=4, ch_blow_out=5)
7
8 app_weld_set_weld_cond_analog(flag_dry_run=1, v_target=200, f_target=150,
9 vel_target=10, vel_min=10,
10 vel_max=100, weld_proc_param=[0.5,0.3,2,1,0.7,0.4,0.7,0.6,1.5])
11
12 app_weld_weave_cond_trapezoidal(wv_offset=[0,0], wv_ang=0, wv_param=[0,5,0,-
13 .7,0.2,0.5,0.7,0.2,0.5])
# 사다리꼴 위빙파형, 옵셋=0,0 기울임각=0, 위빙점1=(0,5), 위빙점2=(0,-5), 위빙 시간
=0.7(sec) 양방향동일, 위빙가감속시간=0.2(sec) 양방향동일, 위빙점1 드웰시간=0.5sec, 위
14 빙점2 드웰시간=0.5sec
15 app_weld_disable_analog ()

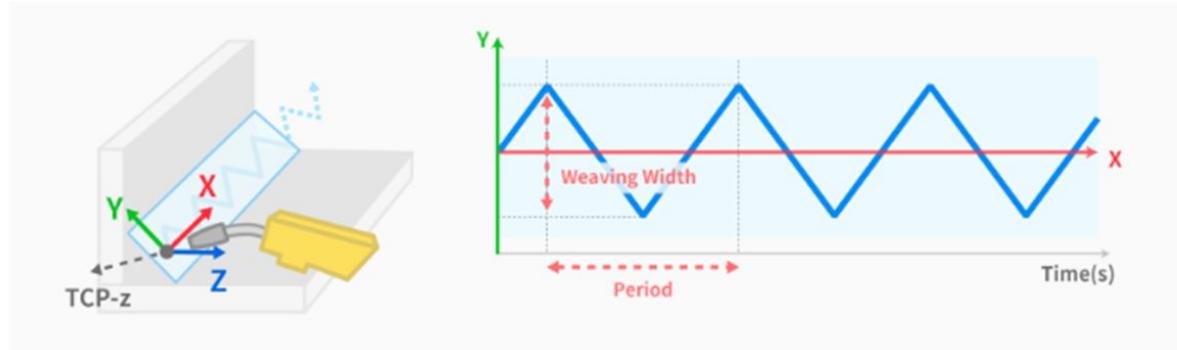
```

### 12.3.20 app\_weld\_weave\_cond\_zigzag(wv\_offset=[0,0], wv\_ang=0, wv\_param=[3,0.6])

## 기능

지그재그 위빙조건을 설정합니다. 위빙조건은 용접기능 활성화 (app\_weld\_enable\_analog()) / app\_weld\_enable\_digital()부터 비활성화(app\_weld\_disable\_analog()/app\_weld\_disable\_digital())로 정의한 용접구간 내에서만 유효하며 이외에 실행 시 에러가 발생합니다. 위빙조건은 용접경로의 진행방향을 위빙x축으로, 위빙x축에서 TCP-z방향을 벡터곱(cross-product)한 방향을 위빙y축으로 하는 위빙좌표계에서 정의됩니다. 좌표계 및 위빙 설정 인자는 아래 그림을 참조하세요. 하나의 용접구간 내에서는 하나의 위빙조건만 허용되며 용접 중 app\_weld\_adj\_welding\_cond\_analog() / app\_weld\_set\_weld\_cond\_digital() 명령어를 통해 옵셋 또는 위빙폭을 조정하거나, 티치펜던트의 용접조건 조정 popup에서 (전압/전류/속도 및) 옵셋을 조정할 수 있습니다. 다만 티치펜던트에서의 용접조건 조정은 명령어를 통한 용접조건 조정상태가 RESET상태 (즉,

app\_weld\_set\_weld\_cond\_analog() / app\_weld\_set\_weld\_cond\_digital()으로 지정한 용접조건 설정) 인 경우에만 가능합니다.



## 인수

인수명	자료형	기본값	설명
wv_offset	float[2]	0	위빙좌표계-y방향 옵셋 (mm)
		0	위빙좌표계-z방향 옵셋 (mm)
wv_angle	float	0	위빙좌표계-x축기준 위빙평면 회전각 (deg)
wv_param	list(float[2])	3	위빙 폭(mm)
		0.6	위빙 주기(sec)

## 리턴

값	설명
0	설정 성공
음수값	설정 실패

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시

예외	설명
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

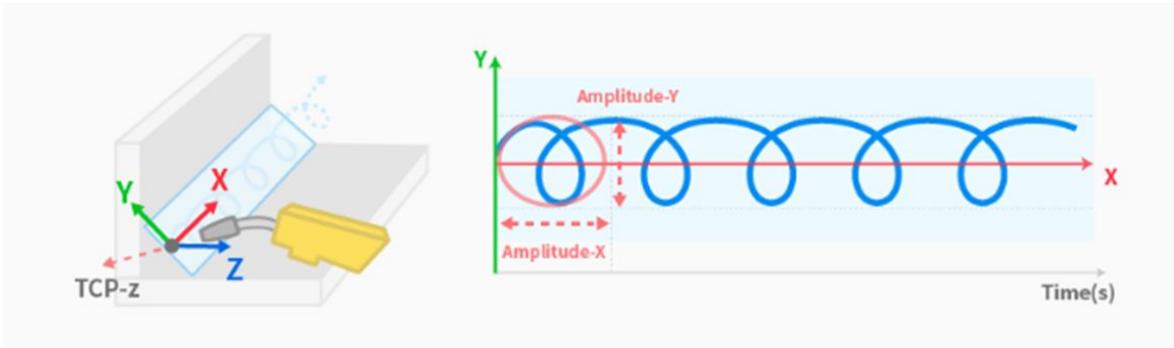
1 app_weld_enable_analog(ch_v_out=[1,1], spec_v_out=[0,0,300,10], ch_f_out =[2
2 ,
3 spec_f_out =[0,0,40,10], ch_v_in =[1,1], spec_v_in =[0,0,300,10], ch_c_in
4 = [2,1],
5 spec_c_in=[0,0,40,10], ch_arc_on=1, ch_gas_on=2, ch_inching_fwd=3,
6 ch_inching_bwd=4, ch_blow_out=5)
7
8 app_weld_set_weld_cond_analog(flag_dry_run=1, v_target=200, f_target=150,
9 vel_target=10, vel_min=10,
10 vel_max=100, weld_proc_param=[0.5,0.3,2,1,0.7,0.4,0.7,0.6,1.5])
11
12 app_weld_weave_cond_zigzag(wv_offset=[0,0], wv_ang=0, wv_param=[10,0.5])
# 지그재그 위빙파형, 옵셋=0,0 기울임각=0, 위빙 폭=10(mm), 위빙 주기=0.5(sec)
13 app_weld_disable_analog()

```

### 12.3.21 app\_weld\_weave\_cond\_circular(wv\_offset=[0,0], wv\_ang=0, wv\_param=[3,3,0.3,0.3])

## 기능

원형 위빙조건을 설정합니다. 위빙조건은 용접기능 활성화 (app\_weld\_enable\_analog()) / app\_weld\_enable\_digital()부터 비활성화(app\_weld\_disable\_analog() / app\_weld\_disable\_digital())로 정의한 용접구간 내에서만 유효하며 이외에 실행 시 에러가 발생합니다. 위빙조건은 용접경로의 진행방향을 위빙x축으로, 위빙x축에서 TCP-z방향을 벡터곱(cross-product)한 방향을 위빙y축으로 하는 위빙좌표계에서 정의됩니다. 좌표계 및 위빙 설정 인자는 아래 그림을 참조하세요. 하나의 용접구간 내에서는 하나의 위빙조건만 허용되며 용접 중 app\_weld\_adj\_welding\_cond\_analog() / app\_weld\_set\_weld\_cond\_digital() 명령어를 통해 옵셋 또는 위빙폭을 조정하거나, 티치펜던트의 용접조건 조정 popup에서 (전압/전류/속도 및) 옵셋을 조정할 수 있습니다. 다만 티치펜던트에서의 용접조건 조정은 명령어를 통한 용접조건 조정상태가 RESET상태 (즉, app\_weld\_set\_weld\_cond\_analog() / app\_weld\_set\_weld\_cond\_digital()으로 지정한 용접조건 설정) 인 경우에만 가능합니다.



## 인수

인수명	자료형	기본값	설명
wv_offset	float[2]	0	위빙좌표계-y방향 옵셋 (mm)
		0	위빙좌표계-z방향 옵셋 (mm)
wv_angle	float	0	위빙좌표계-x축기준 위빙평면 회전각 (deg)
wv_param	list(float[4])	3	x방향 위빙 폭(mm)
		3	y방향 위빙 폭(mm)
		0.3	x방향 위빙 주기(sec)
		0.3	y방향 위빙 주기(sec)

## 리턴

값	설명
0	설정 성공
음수값	설정 실패

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시

예외	설명
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIME)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 app_weld_enable_analog(ch_v_out=[1,1], spec_v_out=[0,0,300,10], ch_f_out =[2
2 ,
3 spec_f_out =[0,0,40,10], ch_v_in =[1,1], spec_v_in =[0,0,300,10], ch_c_in
4 = [2,1],
5 spec_c_in=[0,0,40,10], ch_arc_on=1, ch_gas_on=2, ch_inching_fwd=3,
6 ch_inching_bwd=4, ch_blow_out=5)
7
8 app_weld_set_weld_cond_analog(flag_dry_run=1, v_target=200, f_target=150,
9 vel_target=10, vel_min=10,
10 vel_max=100, weld_proc_param=[0.5,0.3,2,1,0.7,0.4,0.7,0.6,1.5])
11
12 app_weld_weave_cond_circular(wv_offset=[0,0], wv_ang=0, wv_param=[3,3,0.3,0
13 .3])
# 원형 위빙파형, 옵셋=0,0 기울임각=0, x방향위빙폭=3(mm), y방향위빙폭=3(mm), x방향위빙주
14 기=0.3(s), y방향위빙주기=0.3(s)
15 app_weld_disable_analog()

```

### 12.3.22 app\_weld\_weave\_cond\_sinusoidal(wv\_offset=[0,0], wv\_ang=0, wv\_param=[3,0.6])

## 기능

사인 위빙조건을 설정합니다. 위빙조건은 용접기능 활성화 (app\_weld\_enable\_analog()) / app\_weld\_enable\_digital() 부터 비활성화(app\_weld\_disable\_analog() / app\_weld\_disable\_digital())로 정의한 용접구간 내에서만 유효하며 이외에 실행 시 에러가 발생합니다. 위빙조건은 용접경로의 진행방향을 위빙x축으로, 위빙x축에서 TCP-z방향을 벡터곱(cross-product)한 방향을 위빙y축으로 하는 위빙좌표계에서 정의됩니다. 좌표계 및 위빙 설정 인자는 아래 그림을 참조하세요. 하나의 용접구간 내에서는 하나의 위빙조건만 허용되며 용접 중 app\_weld\_adj\_welding\_cond\_analog() / app\_weld\_set\_weld\_cond\_digital() 명령어를 통해 옵셋 또는 위빙폭을 조정하거나, 티치펜던트의 용접조건 조정 popup에서 (전압/전류/속도 및) 옵셋을 조정할 수 있습니다. 다만 티치펜던트에서의 용접조건 조정은 명령어를 통한 용접조건 조정상태가 RESET상태 (즉,

app\_weld\_set\_weld\_cond\_analog()/app\_weld\_set\_weld\_cond\_digital()으로 지정한 용접조건 설정) 인 경우에만 가능합니다.



## 인수

인수명	자료형	기본값	설명
wv_offset	float[2]	0	위빙좌표계-y방향 옵셋 (mm)
		0	위빙좌표계-z방향 옵셋 (mm)
wv_angle	float	0	위빙좌표계-x축기준 위빙평면 회전각 (deg)
wv_param	list(float[2])	3	위빙 폭(mm)
		0.6	위빙 주기(sec)

## 리턴

값	설명
0	설정 성공
음수값	설정 실패

## 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시

예외	설명
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

## 예제

```

1 app_weld_enable_analog(ch_v_out=[1,1], spec_v_out=[0,0,300,10], ch_f_out =[2
2 ,
3 spec_f_out =[0,0,40,10], ch_v_in =[1,1], spec_v_in =[0,0,300,10], ch_c_in
4 = [2,1],
5 spec_c_in=[0,0,40,10], ch_arc_on=1, ch_gas_on=2, ch_inching_fwd=3,
6 ch_inching_bwd=4, ch_blow_out=5)
7
8 app_weld_set_weld_cond_analog(flag_dry_run=1, v_target=200, f_target=150,
9 vel_target=10, vel_min=10,
10 vel_max=100, weld_proc_param=[0.5,0.3,2,1,0.7,0.4,0.7,0.6,1.5])
# 사인 위빙파형, 옵셋=0,0 기울임각=0, 위빙폭=10(mm), 위빙주기=0.5(s)
app_weld_disable_analog()

```

## 13 A-Series 전용 명령어

### 13.1 컨트롤러

#### 13.1.1 get\_function\_input(index)

##### 기능

컨트롤러의 디지털 입력 점접에서 신호를 불러오기 위한 명령문으로 디지털 입력 점접 값을 읽습니다.

Process Button 장치의 Function 버튼 상태를 읽기 위한 명령문입니다.

##### 인수

인수명	자료형	기본값	설명
index	int	-	Process Button 장치의 Function 버튼 번호이며, 1 ~ 4까지 가능함.

##### 리턴

값	설명
1	ON
0	OFF
음수값	실패

##### 예외

예외	설명
DR_Error (DR_ERROR_TYPE)	인수들의 데이터형 오류 시
DR_Error (DR_ERROR_VALUE)	인수의 값이 유효하지 않을 시
DR_Error (DR_ERROR_RUNTIM E)	C Extension 모듈 에러 발생 시

예외	설명
DR_Error (DR_ERROR_STOP)	프로그램 강제 종료 시

### 예제

1	in1 = get_function_input(1) #1번 Function 버튼 상태 읽기
2	in8 = get_function_input(4) #4번 Function 버튼 상태 읽기