

Tennix Ball Assistant [[Group #4]] Kevin Albertson, Pawel Szczurko, Matthew Reyes, Sewon Chung, Edward Choi, Timothy Yong

Generated by Doxygen 1.8.6

Sun Mar 29 2015 06:12:49

Contents

1	Namespace Index	1
1.1	Namespace List	1
2	Hierarchical Index	3
2.1	Class Hierarchy	3
3	Class Index	5
3.1	Class List	5
4	File Index	7
4.1	File List	7
5	Namespace Documentation	9
5.1	basket Namespace Reference	9
5.1.1	Function Documentation	9
5.1.1.1	_basket_image_hue_filter	9
5.1.1.2	_init_particle_filter	9
5.1.1.3	_save_image	10
5.1.1.4	is_basket_middle	10
5.1.1.5	run	10
5.1.1.6	run_middle	10
5.1.2	Variable Documentation	10
5.1.2.1	base_filename	10
5.1.2.2	image_half_size	10
5.1.2.3	particle_filter	10
5.1.2.4	save_count	10
5.2	basket_runner Namespace Reference	10
5.3	basket_test Namespace Reference	10
5.3.1	Function Documentation	11
5.3.1.1	basketMissing	11
5.3.1.2	basketPresent	11
5.3.1.3	unitTest	11
5.4	experiment Namespace Reference	11

5.4.1	Detailed Description	11
5.4.2	Function Documentation	11
5.4.2.1	binary_mask	11
5.4.2.2	blobs_by_mask	11
5.4.2.3	dilation_and_blur	11
5.4.2.4	experiment	11
5.4.2.5	hard_threshold	11
5.5	image_support Namespace Reference	11
5.5.1	Function Documentation	12
5.5.1.1	external_init_particle_filter	12
5.5.1.2	get_best_blob	13
5.5.1.3	get_hue_blobs	13
5.5.1.4	image_hue_filter	13
5.5.1.5	is_blob_in_middle_helper	13
5.6	particlefilter Namespace Reference	14
5.7	prquadtree Namespace Reference	14
5.7.1	Detailed Description	14
5.8	prquadtree_test Namespace Reference	14
5.9	prquadtree_test_example Namespace Reference	14
5.9.1	Detailed Description	15
5.9.2	Variable Documentation	15
5.9.2.1	b	15
5.9.2.2	b2	15
5.9.2.3	c	15
5.9.2.4	nearby	15
5.9.2.5	pt	15
5.9.2.6	qt	15
5.10	tennis_ball Namespace Reference	15
5.10.1	Detailed Description	15
5.10.2	Function Documentation	15
5.10.2.1	_ball_image_hue_filter	15
5.10.2.2	_init_particle_filter	16
5.10.2.3	is_ball_middle	16
5.10.2.4	run	16
5.10.3	Variable Documentation	16
5.10.3.1	particle_filter	16
5.11	tennis_ball_runner Namespace Reference	16
5.12	tennis_ball_test Namespace Reference	16
5.12.1	Function Documentation	16
5.12.1.1	ballMissing	16

5.12.1.2	ballPresent	16
5.12.1.3	unitTest	17
6	Class Documentation	19
6.1	_ctrl Struct Reference	19
6.2	prquadtree.Box Class Reference	19
6.2.1	Detailed Description	19
6.2.2	Constructor & Destructor Documentation	19
6.2.2.1	__init__	19
6.2.3	Member Function Documentation	20
6.2.3.1	contains_point	20
6.2.3.2	intersect	20
6.2.4	Member Data Documentation	20
6.2.4.1	center	20
6.2.4.2	half_size	20
6.3	controller Struct Reference	20
6.3.1	Member Data Documentation	21
6.3.1.1	A	21
6.3.1.2	axes	21
6.3.1.3	B	21
6.3.1.4	buttons	21
6.3.1.5	connected	21
6.3.1.6	DOWN	21
6.3.1.7	fd	21
6.3.1.8	HOME	21
6.3.1.9	LB	21
6.3.1.10	LB2	21
6.3.1.11	LEFT	21
6.3.1.12	LJOY	21
6.3.1.13	LT	21
6.3.1.14	name	22
6.3.1.15	pressed	22
6.3.1.16	RB	22
6.3.1.17	RB2	22
6.3.1.18	RIGHT	22
6.3.1.19	RJOY	22
6.3.1.20	RT	22
6.3.1.21	SELECT	22
6.3.1.22	START	22
6.3.1.23	UP	22

6.3.1.24	X	22
6.3.1.25	x	22
6.3.1.26	Y	22
6.3.1.27	y	22
6.4	HBridgeMotor Class Reference	22
6.4.1	Constructor & Destructor Documentation	22
6.4.1.1	HBridgeMotor	22
6.4.2	Member Function Documentation	22
6.4.2.1	attach	23
6.4.2.2	reset	23
6.4.2.3	setdigital	23
6.4.2.4	write	23
6.4.3	Member Data Documentation	23
6.4.3.1	isdigital	23
6.4.3.2	pin	23
6.4.3.3	velocity	23
6.5	htplink Struct Reference	23
6.5.1	Member Data Documentation	23
6.5.1.1	connected	23
6.5.1.2	hostname	23
6.5.1.3	ipaddr	23
6.5.1.4	socket_fd	23
6.6	info Struct Reference	23
6.6.1	Member Data Documentation	24
6.6.1.1	ss	24
6.7	note Union Reference	24
6.7.1	Member Data Documentation	24
6.7.1.1	"@2	24
6.7.1.2	high	24
6.7.1.3	low	24
6.7.1.4	val	24
6.8	prquadtree.Particle Class Reference	24
6.8.1	Detailed Description	25
6.8.2	Constructor & Destructor Documentation	25
6.8.2.1	__init__	25
6.8.3	Member Data Documentation	25
6.8.3.1	score	25
6.8.3.2	x	25
6.8.3.3	y	25
6.9	particlefilter.ParticleFilter Class Reference	25

6.9.1	Constructor & Destructor Documentation	26
6.9.1.1	__init__	26
6.9.2	Member Function Documentation	26
6.9.2.1	clear_scores	26
6.9.2.2	iterate	26
6.9.2.3	score	26
6.9.3	Member Data Documentation	26
6.9.3.1	image_box	26
6.9.3.2	iterations	26
6.9.3.3	iterations_before_clearing	26
6.9.3.4	pr_tree	26
6.10	prquadtree.Point Class Reference	26
6.10.1	Detailed Description	27
6.10.2	Constructor & Destructor Documentation	27
6.10.2.1	__init__	27
6.10.3	Member Function Documentation	27
6.10.3.1	__repr__	27
6.10.3.2	__str__	27
6.10.4	Member Data Documentation	28
6.10.4.1	x	28
6.10.4.2	y	28
6.11	point2d Struct Reference	28
6.11.1	Member Data Documentation	28
6.11.1.1	x	28
6.11.1.2	y	28
6.12	point3d Struct Reference	28
6.12.1	Member Data Documentation	28
6.12.1.1	x	28
6.12.1.2	y	28
6.12.1.3	z	28
6.13	pose2d Struct Reference	29
6.13.1	Member Data Documentation	29
6.13.1.1	theta	29
6.13.1.2	x	29
6.13.1.3	y	29
6.14	pose3d Struct Reference	29
6.14.1	Member Data Documentation	29
6.14.1.1	pitch	29
6.14.1.2	roll	29
6.14.1.3	x	29

6.14.1.4	y	29
6.14.1.5	yaw	29
6.14.1.6	z	29
6.15	prquadtree.PRQuadTree Class Reference	30
6.15.1	Detailed Description	30
6.15.2	Constructor & Destructor Documentation	30
6.15.2.1	__init__	30
6.15.3	Member Function Documentation	31
6.15.3.1	__str__	31
6.15.3.2	_subdivide	31
6.15.3.3	insert	31
6.15.3.4	print_all_points	31
6.15.3.5	query_k_nearest	31
6.15.3.6	query_range	32
6.15.3.7	size	32
6.15.4	Member Data Documentation	32
6.15.4.1	box	32
6.15.4.2	ne	32
6.15.4.3	nw	32
6.15.4.4	points	32
6.15.4.5	QT_NODE_CAPACITY	32
6.15.4.6	se	32
6.15.4.7	sw	33
6.16	rawrec Struct Reference	33
6.16.1	Member Data Documentation	33
6.16.1.1	buffer	33
6.16.1.2	caps	33
6.16.1.3	filesink_loc	33
6.16.1.4	pipeline	33
6.16.1.5	sink	33
6.16.1.6	source	33
6.17	serial Struct Reference	33
6.17.1	Member Data Documentation	34
6.17.1.1	baudrate	34
6.17.1.2	buffer	34
6.17.1.3	connected	34
6.17.1.4	fd	34
6.17.1.5	parity	34
6.17.1.6	port	34
6.17.1.7	readAvailable	34

6.17.1.8 readbuf	34
6.18 speech_signal Struct Reference	34
6.18.1 Member Data Documentation	34
6.18.1.1 fetch	34
6.18.1.2 go	34
6.18.1.3 none	34
6.18.1.4 ret	34
6.18.1.5 stop	34
6.19 stt Struct Reference	34
6.19.1 Member Data Documentation	35
6.19.1.1 config	35
6.19.1.2 nothing	35
6.19.1.3 ps	35
6.20 tbr Struct Reference	35
6.20.1 Member Data Documentation	35
6.20.1.1 arm	35
6.20.1.2 claw	35
6.20.1.3 connected	35
6.20.1.4 connections	35
6.20.1.5 ids	35
6.20.1.6 left	35
6.20.1.7 num_possible	36
6.20.1.8 possible_ports	36
6.20.1.9 prev_arm	36
6.20.1.10 prev_claw	36
6.20.1.11 prev_left	36
6.20.1.12 prev_right	36
6.20.1.13 right	36
6.21 prquadtree_test.TestBox Class Reference	36
6.21.1 Member Function Documentation	36
6.21.1.1 test_box_contains	36
6.21.1.2 test_box_insert	36
6.22 prquadtree_test.TestParticle Class Reference	36
6.22.1 Member Function Documentation	37
6.22.1.1 test_particle_insert	37
6.23 prquadtree_test.TestPoint Class Reference	37
6.23.1 Member Function Documentation	37
6.23.1.1 test_point_insert	37
6.24 prquadtree_test.TestPrQuadTree Class Reference	37
6.24.1 Member Function Documentation	38

6.24.1.1	test_insert	38
6.24.1.2	test_nearby	38
7	File Documentation	39
7.1	SEVATBR-doc/agent/agent.c File Reference	39
7.1.1	Typedef Documentation	39
7.1.1.1	info_t	39
7.1.2	Function Documentation	39
7.1.2.1	conscious_thought	40
7.1.2.2	gotosleep	41
7.1.2.3	wakeup	41
7.1.3	Variable Documentation	41
7.1.3.1	consciousness	41
7.1.3.2	ctrl	41
7.1.3.3	end_consciousness	41
7.2	SEVATBR-doc/agent/agent.h File Reference	41
7.2.1	Macro Definition Documentation	41
7.2.1.1	AGENT_SIMPLE	41
7.2.2	Function Documentation	41
7.2.2.1	agent_create	41
7.2.2.2	agent_destroy	42
7.2.2.3	agent_disable	42
7.2.2.4	agent_enable	42
7.2.2.5	agent_get_poses	42
7.3	SEVATBR-doc/agent/simple_agent.cpp File Reference	42
7.3.1	Function Documentation	43
7.3.1.1	agent_create	43
7.3.1.2	agent_destroy	44
7.3.1.3	agent_disable	44
7.3.1.4	agent_enable	44
7.3.1.5	agent_get_poses	44
7.3.2	Variable Documentation	44
7.3.2.1	agent_enabled	44
7.3.2.2	last_signal	44
7.3.2.3	view	44
7.4	SEVATBR-doc/agent/test.c File Reference	44
7.4.1	Function Documentation	45
7.4.1.1	main	45
7.4.1.2	stopprog	45
7.4.2	Variable Documentation	45

7.4.2.1	exit_signal	45
7.5	SEVATBR-doc/manual/test.c File Reference	45
7.5.1	Function Documentation	45
7.5.1.1	main	45
7.5.1.2	stop	45
7.5.2	Variable Documentation	45
7.5.2.1	stopsig	45
7.6	SEVATBR-doc/robot/tbr/test.c File Reference	45
7.6.1	Function Documentation	45
7.6.1.1	main	45
7.7	SEVATBR-doc/speech/speech-to-text/test.c File Reference	46
7.7.1	Function Documentation	46
7.7.1.1	main	46
7.7.1.2	stop	46
7.7.2	Variable Documentation	46
7.7.2.1	stopsig	46
7.8	SEVATBR-doc/core/coord.h File Reference	46
7.8.1	Typedef Documentation	46
7.8.1.1	point2d_t	46
7.8.1.2	point3d_t	46
7.8.1.3	point_t	46
7.8.1.4	pose2d_t	47
7.8.1.5	pose3d_t	47
7.8.1.6	pose_t	47
7.9	SEVATBR-doc/core/core.c File Reference	47
7.9.1	Function Documentation	47
7.9.1.1	main	47
7.9.1.2	stop_program	47
7.9.2	Variable Documentation	47
7.9.2.1	stop_signal	47
7.10	SEVATBR-doc/core/core.cpp File Reference	48
7.10.1	Function Documentation	48
7.10.1.1	main	48
7.10.1.2	stop_program	48
7.10.2	Variable Documentation	48
7.10.2.1	stop_signal	48
7.11	SEVATBR-doc/logger/logger.c File Reference	48
7.11.1	Function Documentation	49
7.11.1.1	logger_destroy	49
7.11.1.2	logger_init	49

7.11.1.3	logger_print	49
7.11.1.4	logger_scan	49
7.11.2	Variable Documentation	49
7.11.2.1	logfile	49
7.12	SEVATBR-doc/logger/logger.h File Reference	49
7.12.1	Function Documentation	50
7.12.1.1	log_destroy	50
7.12.1.2	log_init	50
7.12.1.3	log_print	50
7.12.1.4	log_scan	50
7.13	SEVATBR-doc/manual/controller.c File Reference	50
7.13.1	Macro Definition Documentation	50
7.13.1.1	INPUT_DIR	50
7.13.1.2	JS_PREFIX	50
7.13.1.3	MAX_16BIT	50
7.13.2	Function Documentation	50
7.13.2.1	controller_connect	50
7.13.2.2	controller_disconnect	51
7.13.2.3	controller_update	52
7.14	SEVATBR-doc/manual/controller.h File Reference	52
7.14.1	Typedef Documentation	52
7.14.1.1	controller_t	52
7.14.2	Function Documentation	52
7.14.2.1	controller_connect	52
7.14.2.2	controller_disconnect	52
7.15	SEVATBR-doc/manual/httplink.c File Reference	53
7.15.1	Function Documentation	53
7.15.1.1	httplink_connect	53
7.15.1.2	httplink_disconnect	53
7.15.1.3	httplink_recv	53
7.15.1.4	httplink_send	54
7.15.2	Variable Documentation	54
7.15.2.1	msgbuf	54
7.15.2.2	response	54
7.16	SEVATBR-doc/manual/httplink.h File Reference	54
7.16.1	Typedef Documentation	54
7.16.1.1	httplink_t	54
7.16.2	Function Documentation	54
7.16.2.1	httplink_connect	54
7.16.2.2	httplink_disconnect	55

7.16.2.3	httplink_recv	55
7.16.2.4	httplink_send	55
7.17	SEVATBR-doc/manual/manual.c File Reference	55
7.17.1	Macro Definition Documentation	56
7.17.1.1	HZ	56
7.17.2	Function Documentation	56
7.17.2.1	controller_update	56
7.17.2.2	isOverriden	56
7.17.2.3	manual_connect	56
7.17.2.4	manual_disable	57
7.17.2.5	manual_disconnect	57
7.17.2.6	manual_enable	57
7.17.2.7	manual_get_poses	57
7.17.2.8	manual_new_data	57
7.17.2.9	raise_server_request	57
7.17.2.10	server_update	57
7.17.3	Variable Documentation	58
7.17.3.1	arm	58
7.17.3.2	base	58
7.17.3.3	ctrl	58
7.17.3.4	input_id	58
7.17.3.5	last_signal	58
7.17.3.6	manual_en	58
7.17.3.7	mnl_override	58
7.17.3.8	new_join	58
7.17.3.9	server	58
7.18	SEVATBR-doc/manual/manual.h File Reference	58
7.18.1	Macro Definition Documentation	58
7.18.1.1	MNL_CTRL	58
7.18.1.2	MNL_SRVR	58
7.18.2	Function Documentation	58
7.18.2.1	isOverriden	58
7.18.2.2	manual_connect	59
7.18.2.3	manual_disable	59
7.18.2.4	manual_disconnect	59
7.18.2.5	manual_enable	59
7.18.2.6	manual_get_poses	59
7.18.2.7	manual_new_data	59
7.19	SEVATBR-doc/manual/rtsp/rtspink.c File Reference	59
7.20	SEVATBR-doc/robot/arduino/arm/arm.cpp File Reference	60

7.20.1	Macro Definition Documentation	60
7.20.1.1	ARM_L1	60
7.20.1.2	ARM_L2	60
7.20.1.3	ARM_L3	60
7.20.1.4	ARM_R1	60
7.20.1.5	ARM_R2	60
7.20.1.6	ARM_R3	60
7.20.1.7	DEV_ID	60
7.20.2	Function Documentation	60
7.20.2.1	limit	60
7.20.2.2	loop	61
7.20.2.3	lowerarm	61
7.20.2.4	raisearm	61
7.20.2.5	setarm	61
7.20.2.6	setup	61
7.20.2.7	stoparm	61
7.20.3	Variable Documentation	61
7.20.3.1	arm_l	61
7.20.3.2	arm_r	61
7.20.3.3	msg	61
7.20.3.4	time	61
7.21	SEVATBR-doc/robot/arduino/claw/claw.cpp File Reference	61
7.21.1	Macro Definition Documentation	62
7.21.1.1	BOT_L	62
7.21.1.2	BOT_R	62
7.21.1.3	CLAW_L	62
7.21.1.4	CLAW_R	62
7.21.1.5	DEV_ID	62
7.21.1.6	TOP_L	62
7.21.1.7	TOP_R	62
7.21.2	Function Documentation	62
7.21.2.1	closeclaw	62
7.21.2.2	limit	62
7.21.2.3	loop	62
7.21.2.4	openclaw	62
7.21.2.5	setclaw	62
7.21.2.6	setup	62
7.21.2.7	stopclaw	62
7.21.3	Variable Documentation	62
7.21.3.1	claw_l	62

7.21.3.2	claw_r	62
7.21.3.3	msg	62
7.21.3.4	time	62
7.22	SEVATBR-doc/robot/arduino/wheels/wheels.cpp File Reference	62
7.22.1	Macro Definition Documentation	63
7.22.1.1	DEV_ID	63
7.22.1.2	WHEEL_LB1	63
7.22.1.3	WHEEL_LB2	63
7.22.1.4	WHEEL_LM1	63
7.22.1.5	WHEEL_LM2	63
7.22.1.6	WHEEL_LT1	63
7.22.1.7	WHEEL_LT2	63
7.22.1.8	WHEEL_RB1	63
7.22.1.9	WHEEL_RB2	63
7.22.1.10	WHEEL_RM1	63
7.22.1.11	WHEEL_RM2	64
7.22.1.12	WHEEL_RT1	64
7.22.1.13	WHEEL_RT2	64
7.22.2	Function Documentation	64
7.22.2.1	backward	64
7.22.2.2	forward	64
7.22.2.3	limit	64
7.22.2.4	loop	64
7.22.2.5	setup	64
7.22.2.6	setwheels	64
7.22.2.7	stopwheels	64
7.22.2.8	turnleft	64
7.22.2.9	turnright	64
7.22.3	Variable Documentation	64
7.22.3.1	msg	64
7.22.3.2	time	64
7.22.3.3	wheel_l	64
7.22.3.4	wheel_r	64
7.23	SEVATBR-doc/robot/robot.c File Reference	64
7.23.1	Function Documentation	65
7.23.1.1	robot_move	65
7.23.1.2	robot_set	65
7.23.1.3	robot_unset	65
7.23.2	Variable Documentation	65
7.23.2.1	currid	65

7.23.2.2	robot	65
7.24	SEVATBR-doc/robot/robot.h File Reference	65
7.24.1	Macro Definition Documentation	66
7.24.1.1	NO_ROBOT	66
7.24.1.2	STANDARD_OUT	66
7.24.1.3	TACHIKOMA	66
7.24.1.4	TENNIS_BALL_ROBOT	66
7.24.2	Function Documentation	66
7.24.2.1	robot_move	66
7.24.2.2	robot_set	66
7.24.2.3	robot_unset	66
7.25	SEVATBR-doc/robot/serial/serial.c File Reference	67
7.25.1	Macro Definition Documentation	67
7.25.1.1	INPUT_DIR	67
7.25.2	Function Documentation	67
7.25.2.1	_serial_setattr	67
7.25.2.2	_serial_update	67
7.25.2.3	serial_connect	68
7.25.2.4	serial_disconnect	68
7.25.2.5	serial_read	68
7.25.2.6	serial_write	68
7.25.3	Variable Documentation	69
7.25.3.1	PREFIXES	69
7.25.3.2	tempbuf	69
7.26	SEVATBR-doc/robot/serial/serial.h File Reference	69
7.26.1	Macro Definition Documentation	69
7.26.1.1	SWBUFMAX	69
7.26.1.2	SWREADMAX	69
7.26.1.3	SWWRITEMAX	69
7.26.2	Typedef Documentation	69
7.26.2.1	serial_t	69
7.26.3	Function Documentation	70
7.26.3.1	serial_connect	70
7.26.3.2	serial_disconnect	71
7.26.3.3	serial_read	71
7.26.3.4	serial_write	71
7.27	SEVATBR-doc/robot/tbr/tbr.c File Reference	71
7.27.1	Macro Definition Documentation	72
7.27.1.1	ARM_DEVID	72
7.27.1.2	CLAW_DEVID	72

7.27.1.3	DEV_BAUD	72
7.27.1.4	NUM_DEV	72
7.27.1.5	SYNC_NSEC	72
7.27.1.6	WHEEL_DEVID	72
7.27.2	Function Documentation	72
7.27.2.1	tbr_connect	72
7.27.2.2	tbr_disconnect	72
7.27.2.3	tbr_recv	73
7.27.2.4	tbr_reset	73
7.27.2.5	tbr_send	73
7.28	SEVATBR-doc/robot/tbr/tbr.h File Reference	73
7.28.1	Typedef Documentation	74
7.28.1.1	tbr_t	74
7.28.2	Function Documentation	74
7.28.2.1	tbr_connect	74
7.28.2.2	tbr_disconnect	74
7.28.2.3	tbr_recv	74
7.28.2.4	tbr_reset	74
7.28.2.5	tbr_send	74
7.29	SEVATBR-doc/slam/slam.c File Reference	74
7.29.1	Function Documentation	75
7.29.1.1	gauss_error	75
7.29.1.2	move	75
7.29.1.3	particle_resample	75
7.29.1.4	particle_update	75
7.29.1.5	sense	75
7.29.1.6	set_noise	75
7.30	SEVATBR-doc/speech/speech-to-text/lse_stt.c File Reference	75
7.30.1	Macro Definition Documentation	76
7.30.1.1	NTARGETS	76
7.30.2	Function Documentation	76
7.30.2.1	loadFile	76
7.30.2.2	square_error	76
7.30.2.3	stt_decipher	76
7.30.2.4	stt_free	77
7.30.2.5	stt_init	78
7.30.3	Variable Documentation	78
7.30.3.1	db	78
7.30.3.2	dbfiles	78
7.30.3.3	goodtogo	78

7.30.3.4	length	78
7.30.3.5	text	78
7.31	SEVATBR-doc/speech/speech-to-text/lse_stt.h File Reference	78
7.31.1	Typedef Documentation	78
7.31.1.1	stt_t	78
7.31.2	Function Documentation	78
7.31.2.1	stt_decipher	78
7.31.2.2	stt_free	79
7.31.2.3	stt_init	79
7.32	SEVATBR-doc/speech/speech-to-text/rawrec.c File Reference	80
7.32.1	Function Documentation	80
7.32.1.1	start_recording	80
7.32.1.2	stop_recording	80
7.32.2	Variable Documentation	80
7.32.2.1	gst_initialized	80
7.33	SEVATBR-doc/speech/speech-to-text/rawrec.h File Reference	80
7.33.1	Typedef Documentation	81
7.33.1.1	rawrec_t	81
7.33.2	Function Documentation	81
7.33.2.1	start_recording	81
7.33.2.2	stop_recording	81
7.34	SEVATBR-doc/speech/speech-to-text/speech_signal.c File Reference	81
7.34.1	Macro Definition Documentation	82
7.34.1.1	READPROG	82
7.34.2	Function Documentation	82
7.34.2.1	_update_signals	82
7.34.2.2	get_signal	82
7.34.2.3	start_speech_signals	82
7.34.2.4	stop_speech_signals	82
7.34.3	Variable Documentation	83
7.34.3.1	gst_lock	83
7.34.3.2	gst_manager	83
7.34.3.3	gst_sig	83
7.34.3.4	sre	83
7.34.3.5	sslib_exit	83
7.34.3.6	sslib_init	83
7.35	SEVATBR-doc/speech/speech-to-text/speech_signal.h File Reference	83
7.35.1	Typedef Documentation	83
7.35.1.1	speech_signal_t	83
7.35.2	Function Documentation	83

7.35.2.1	get_signal	83
7.35.2.2	start_speech_signals	83
7.35.2.3	stop_speech_signals	83
7.36	SEVATBR-doc/speech/speech-to-text/stt.c File Reference	84
7.36.1	Function Documentation	84
7.36.1.1	stt_decipher	84
7.36.1.2	stt_free	84
7.36.1.3	stt_init	84
7.37	SEVATBR-doc/speech/speech-to-text/stt.h File Reference	84
7.37.1	Typedef Documentation	85
7.37.1.1	stt_t	85
7.37.2	Function Documentation	85
7.37.2.1	stt_decipher	85
7.37.2.2	stt_free	85
7.37.2.3	stt_init	86
7.38	SEVATBR-doc/visual/basket.py File Reference	86
7.39	SEVATBR-doc/visual/basket_runner.py File Reference	87
7.40	SEVATBR-doc/visual/basket_test.py File Reference	87
7.41	SEVATBR-doc/visual/experiment.py File Reference	87
7.42	SEVATBR-doc/visual/image_support.py File Reference	87
7.43	SEVATBR-doc/visual/particlefilter.py File Reference	88
7.44	SEVATBR-doc/visual/prquadtree.py File Reference	88
7.45	SEVATBR-doc/visual/prquadtree_test.py File Reference	88
7.46	SEVATBR-doc/visual/prquadtree_test_example.py File Reference	88
7.47	SEVATBR-doc/visual/tennis_ball.py File Reference	89
7.48	SEVATBR-doc/visual/tennis_ball_runner.py File Reference	89
7.49	SEVATBR-doc/visual/tennis_ball_test.py File Reference	89
7.50	SEVATBR-doc/visual/visual.h File Reference	90
7.50.1	Function Documentation	90
7.50.1.1	get_objects	90
7.50.1.2	set_objects	90
7.50.1.3	start_visual	90
7.50.1.4	stop_visual	90

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

basket	9
basket_runner	10
basket_test	10
experiment	11
image_support	11
particlefilter	14
prquadtree	14
prquadtree_test	14
prquadtree_test_example	14
tennis_ball	15
tennis_ball_runner	16
tennis_ball_test	16

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

_ctrl	19
prquadtree.Box	19
controller	20
HBridgeMotor	22
httplink	23
info	23
note	24
particlefilter.ParticleFilter	25
prquadtree.Point	26
prquadtree.Particle	24
point2d	28
point3d	28
pose2d	29
pose3d	29
prquadtree.PRQuadTree	30
rawrec	33
serial	33
speech_signal	34
stt	34
tbr	35
TestCase	
prquadtree_test.TestBox	36
prquadtree_test.TestParticle	36
prquadtree_test.TestPoint	37
prquadtree_test.TestPrQuadTree	37

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

_ctrl	19
prquadtree.Box	
Class defining a square on the coordinate system via a center point and half of square width	19
controller	20
HBridgeMotor	22
httplink	23
info	23
note	24
prquadtree.Particle	
Represents particle point	24
particlefilter.ParticleFilter	25
prquadtree.Point	
Represents an (x,y) coordinate point on a grid	26
point2d	28
point3d	28
pose2d	29
pose3d	29
prquadtree.PRQuadTree	
Class representing a Point Range Quadtree	30
rawrec	33
serial	33
speech_signal	34
stt	34
tbr	35
prquadtree_test.TestBox	36
prquadtree_test.TestParticle	36
prquadtree_test.TestPoint	37
prquadtree_test.TestPrQuadTree	37

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

SEVATBR-doc/agent/agent.c	39
SEVATBR-doc/agent/agent.h	41
SEVATBR-doc/agent/simple_agent.cpp	42
SEVATBR-doc/agent/test.c	44
SEVATBR-doc/core/coord.h	46
SEVATBR-doc/core/core.c	47
SEVATBR-doc/core/core.cpp	48
SEVATBR-doc/logger/logger.c	48
SEVATBR-doc/logger/logger.h	49
SEVATBR-doc/manual/controller.c	50
SEVATBR-doc/manual/controller.h	52
SEVATBR-doc/manual/httplink.c	53
SEVATBR-doc/manual/httplink.h	54
SEVATBR-doc/manual/manual.c	55
SEVATBR-doc/manual/manual.h	58
SEVATBR-doc/manual/test.c	45
SEVATBR-doc/manual/rtp/rtsplink.c	59
SEVATBR-doc/robot/robot.c	64
SEVATBR-doc/robot/robot.h	65
SEVATBR-doc/robot/arduino/arm/arm.cpp	60
SEVATBR-doc/robot/arduino/claw/claw.cpp	61
SEVATBR-doc/robot/arduino/wheels/wheels.cpp	62
SEVATBR-doc/robot/serial/serial.c	67
SEVATBR-doc/robot/serial/serial.h	69
SEVATBR-doc/robot/tbr/tbr.c	71
SEVATBR-doc/robot/tbr/tbr.h	73
SEVATBR-doc/robot/tbr/test.c	45
SEVATBR-doc/slam/slam.c	74
SEVATBR-doc/speech/speech-to-text/lse_stt.c	75
SEVATBR-doc/speech/speech-to-text/lse_stt.h	78
SEVATBR-doc/speech/speech-to-text/rawrec.c	80
SEVATBR-doc/speech/speech-to-text/rawrec.h	80
SEVATBR-doc/speech/speech-to-text/speech_signal.c	81
SEVATBR-doc/speech/speech-to-text/speech_signal.h	83
SEVATBR-doc/speech/speech-to-text/stt.c	84
SEVATBR-doc/speech/speech-to-text/stt.h	84
SEVATBR-doc/speech/speech-to-text/test.c	46
SEVATBR-doc/visual/basket.py	86

SEVATBR-doc/visual/ basket_runner.py	87
SEVATBR-doc/visual/ basket_test.py	87
SEVATBR-doc/visual/ experiment.py	87
SEVATBR-doc/visual/ image_support.py	87
SEVATBR-doc/visual/ particlefilter.py	88
SEVATBR-doc/visual/ prquadtree.py	88
SEVATBR-doc/visual/ prquadtree_test.py	88
SEVATBR-doc/visual/ prquadtree_test_example.py	88
SEVATBR-doc/visual/ tennis_ball.py	89
SEVATBR-doc/visual/ tennis_ball_runner.py	89
SEVATBR-doc/visual/ tennis_ball_test.py	89
SEVATBR-doc/visual/ visual.h	90

Chapter 5

Namespace Documentation

5.1 basket Namespace Reference

Functions

- `def _basket_image_hue_filter`
Internal wrapper image hue filter.
- `def _save_image`
Saves an image to the current directory.
- `def _init_particle_filter`
Internal wrapper to particle filter initializer.
- `def is_basket_middle`
Single entry function returning True/False if basket is in the middle of the screen.
- `def run_middle`
Runs continuously and prints if the best detected blob is in the middle.
- `def run`
Runs continuously outlines best matched blob if it is in the middle.

Variables

- `particle_filter` = None
- `int image_half_size` = -1
- `int save_count` = 1
- `tuple base_filename` = `datetime.now()`

5.1.1 Function Documentation

5.1.1.1 `def basket._basket_image_hue_filter (img) [private]`

Internal wrapper image hue filter.

param `img` `SimpleCV.Image` the image to apply the hue filter to

5.1.1.2 `def basket._init_particle_filter (img) [private]`

Internal wrapper to particle filter initializer.

Parameters

<i>img</i>	SimpleCV.Image Any image captured from the Camera, used to initialize the size
------------	--

5.1.1.3 `def basket._save_image(img) [private]`

Saves an image to the current directory.

Parameters

<i>img</i>	SimpleCV.Image the image to save
------------	----------------------------------

5.1.1.4 `def basket.is_basket_middle(img)`

Single entry function returning True/False if basket is in the middle of the screen.

Parameters

<i>img</i>	SimpleCV.Image The image to test
------------	----------------------------------

5.1.1.5 `def basket.run(bestBlobCallback = False)`

Runs continuously outlines best matched blob if it is in the middle.

Parameters

<i>bestBlob-Callback</i>	function Callback called passing the best blob found
--------------------------	--

5.1.1.6 `def basket.run_middle()`

Runs continuously and prints if the best detected blob is in the middle.

5.1.2 Variable Documentation

5.1.2.1 `tuple basket.base_filename = datetime.now()`

5.1.2.2 `int basket.image_half_size = -1`

5.1.2.3 `basket.particle_filter = None`

5.1.2.4 `int basket.save_count = 1`

5.2 basket_runner Namespace Reference

5.3 basket_test Namespace Reference

Functions

- `def unitTest`
- `def basketPresent`
- `def basketMissing`

5.3.1 Function Documentation

5.3.1.1 `def basket_test.basketMissing ()`

5.3.1.2 `def basket_test.basketPresent ()`

5.3.1.3 `def basket_test.unitTest (actual, expected, name)`

5.4 experiment Namespace Reference

Functions

- `def experiment`
- `def hard_threshold`
- `def binary_mask`
- `def dilation_and_blur`
- `def blobs_by_mask`

5.4.1 Detailed Description

A utility file for testing out computer vision techniques on preset images. The purpose of this is to avoid using the webcam, and test on consistent test cases.

5.4.2 Function Documentation

5.4.2.1 `def experiment.binary_mask (img)`

5.4.2.2 `def experiment.blobs_by_mask (img)`

5.4.2.3 `def experiment.dilation_and_blur (img)`

5.4.2.4 `def experiment.experiment (image_function = None, blob_function = None, directory = " . / ")`

5.4.2.5 `def experiment.hard_threshold (img)`

5.5 image_support Namespace Reference

Functions

- `def external_init_particle_filter`
Initializes particle filter.
- `def image_hue_filter`
Converts given image to HSV based on the given color.
- `def get_hue_blobs`
Gets basket blobs after hue distance filtering.
- `def get_best_blob`
Returns the best blob out of the provided set and particle filter.
- `def is_blob_in_middle_helper`
Determines whether the given blob is in ceter of image.

5.5.1 Function Documentation

5.5.1.1 `def image_support.external_init_particle_filter (img)`

Initializes particle filter.

Parameters

<i>img</i>	SimpleCV.Image captured image
------------	-------------------------------

Returns

A ParticleFilter object

5.5.1.2 def image_support.get_best_blob (*blobs*, *particle_filter*)

Returns the best blob out of the provided set and particle filter.

Parameters

<i>blobs</i>	list of potential HSV blobs
<i>particle_filter</i>	initialized ParticleFilter object

Returns

The largest blob found or None.

5.5.1.3 def image_support.get_hue_blobs (*img*)

Gets basket blobs after hue distance filtering.

Parameters

<i>img</i>	SimpleCV.Image captured image.
------------	--------------------------------

Returns

Set of 'black' potential blobs.

5.5.1.4 def image_support.image_hue_filter (*img*, *ball* = True)

Converts given image to HSV based on the given color.

Parameters

<i>img</i>	SimpleCV.Image captured image
<i>color</i>	tuple of RGB values of single 'H' value of HSV

Returns

HSV converted image

5.5.1.5 def image_support.is_blob_in_middle_helper (*img*, *blob*)

Determines whether the given blob is in center of image.

Parameters

<i>img</i>	SimpleCV.Image caputed image
<i>blob</i>	SimpleCV.Blob Blob object

Returns

True if blob in middle of image, false otherwise.

5.6 particlefilter Namespace Reference

Classes

- class [ParticleFilter](#)

5.7 prquadtree Namespace Reference

Classes

- class [Point](#)
Represents an (x,y) coordinate point on a grid.
- class [Particle](#)
Represents particle point.
- class [Box](#)
Class defining a square on the coordinate system via a center point and half of square width.
- class [PRQuadTree](#)
Class representing a [Point](#) Range Quadtree.

5.7.1 Detailed Description

Implementation of a Point Range Quadtree.

Author: Pawel Szczurko

5.8 prquadtree_test Namespace Reference

Classes

- class [TestPoint](#)
- class [TestParticle](#)
- class [TestBox](#)
- class [TestPrQuadTree](#)

5.9 prquadtree_test_example Namespace Reference

Variables

- tuple `b` = `Box(Point(5,5), 50)`
- tuple `b2` = `Box(Point(50,50), 50)`
- tuple `qt` = `PRQuadTree(b2)`
- tuple `pt` = `Point(2,2)`
- tuple `nearby` = `qt.query_k_nearest(pt, 20)`
- int `c` = 1

5.9.1 Detailed Description

File testing the capabilities of the PRQuadTree.

Author: Pawel Szczurko

5.9.2 Variable Documentation

5.9.2.1 tuple prquadtree_test_example.b = **Box**(Point(5,5), 50)

5.9.2.2 tuple prquadtree_test_example.b2 = **Box**(Point(50,50), 50)

5.9.2.3 int prquadtree_test_example.c = 1

5.9.2.4 tuple prquadtree_test_example.nearby = qt.query_k_nearest(pt, 20)

5.9.2.5 tuple prquadtree_test_example.pt = **Point**(2,2)

5.9.2.6 tuple prquadtree_test_example.qt = **PRQuadTree**(b2)

5.10 tennis_ball Namespace Reference

Functions

- def [_init_particle_filter](#)
Internal wrapper to particle filter initializer.
- def [_ball_image_hue_filter](#)
Internal wrapper image hue filter.
- def [is_ball_middle](#)
Entry point for module which determines whether tennis ball is in the middle of the image.
- def [run](#)
Continuously captures image from computer camera and feeds it to the is_ball_middle method to detect whether tennis ball is in the middle of the screen.

Variables

- [particle_filter](#) = None

5.10.1 Detailed Description

Simple detection of ball using SimpleCV (much easier than OpenCV). The run method identifies a tennis ball in the camera stream image. 'is_ball_middle' function can be used to determine whether a ball is horizontally centered based on a specified threshold.

-Pawel Szczurko

5.10.2 Function Documentation

5.10.2.1 def tennis_ball._ball_image_hue_filter (*img*) [*private*]

Internal wrapper image hue filter.

Parameters

<i>img</i>	SimpleCV.Image
------------	----------------

Returns

img SimpleCV.Image converted to HSV

5.10.2.2 `def tennis_ball._init_particle_filter (img) [private]`

Internal wrapper to particle filter initializer.

Parameters

<i>img</i>	SimpleCV.Image
------------	----------------

5.10.2.3 `def tennis_ball.is_ball_middle (img)`

Entry point for module which determines whether tennis ball is in the middle of the image.

Parameters

<i>img</i>	SimpleCV.Image
------------	----------------

Returns

boolean. True if tennis ball is in middle, false otherwise.

5.10.2.4 `def tennis_ball.run ()`

Continuously captures image from computer camera and feeds it to the `is_ball_middle` method to detect whether tennis ball is in the middle of the screen.

5.10.3 Variable Documentation

5.10.3.1 `tennis_ball.particle_filter = None`5.11 `tennis_ball_runner` Namespace Reference5.12 `tennis_ball_test` Namespace Reference

Functions

- `def unitTest`
- `def ballPresent`
- `def ballMissing`

5.12.1 Function Documentation

5.12.1.1 `def tennis_ball_test.ballMissing ()`5.12.1.2 `def tennis_ball_test.ballPresent ()`

5.12.1.3 `def tennis_ball_test.unitTest (actual, expected, name)`

Chapter 6

Class Documentation

6.1 `_ctrl` Struct Reference

The documentation for this struct was generated from the following file:

- SEVATBR-doc/agent/[agent.c](#)

6.2 `prquadtree.Box` Class Reference

Class defining a square on the coordinate system via a center point and half of square width.

Public Member Functions

- def [__init__](#)
Construct a [Box](#) object.
- def [contains_point](#)
Verifies that the given point is within this square.
- def [intersect](#)
Checks if the provided box/square intersects with this square.

Public Attributes

- [center](#)
- [half_size](#)

6.2.1 Detailed Description

Class defining a square on the coordinate system via a center point and half of square width.

6.2.2 Constructor & Destructor Documentation

6.2.2.1 `def prquadtree.Box.__init__(self, center, half_size)`

Construct a [Box](#) object.

Parameters

<i>center</i>	Point type specifying the center of the square
<i>half_size</i>	int half the length of the square

6.2.3 Member Function Documentation

6.2.3.1 `def prquadtree.Box.contains_point (self, point)`

Verifies that the given point is within this square.

Parameters

<i>point</i>	Point type to check if it's in the square
--------------	---

Returns

boolean indicating whether the point is within the square

6.2.3.2 `def prquadtree.Box.intersect (self, other_box)`

Checks if the provided box/square intersects with this square.

Parameters

<i>other_box</i>	Box object
------------------	----------------------------

Returns

Boolean indicating if the two intersect anywhere

6.2.4 Member Data Documentation

6.2.4.1 `prquadtree.Box.center`

6.2.4.2 `prquadtree.Box.half_size`

The documentation for this class was generated from the following file:

- SEVATBR-doc/visual/[prquadtree.py](#)

6.3 controller Struct Reference

```
#include <controller.h>
```

Public Attributes

- char * [name](#)
- int32_t [fd](#)
- int8_t [connected](#)
- int32_t [buttons](#)
- int32_t [axes](#)
- int8_t [A](#)
- int8_t [B](#)

- int8_t [X](#)
- int8_t [Y](#)
- int8_t [UP](#)
- int8_t [DOWN](#)
- int8_t [LEFT](#)
- int8_t [RIGHT](#)
- int8_t [LB](#)
- int8_t [RB](#)
- float [LT](#)
- float [RT](#)
- struct {
 - int8_t [pressed](#)
 - float [x](#)
 - float [y](#)
 } [LJOY](#)
- struct {
 - int8_t [pressed](#)
 - float [x](#)
 - float [y](#)
 } [RJOY](#)
- int8_t [START](#)
- int8_t [SELECT](#)
- int8_t [HOME](#)
- int8_t [LB2](#)
- int8_t [RB2](#)

6.3.1 Member Data Documentation

6.3.1.1 int8_t controller::A

6.3.1.2 int32_t controller::axes

6.3.1.3 int8_t controller::B

6.3.1.4 int32_t controller::buttons

6.3.1.5 int8_t controller::connected

6.3.1.6 int8_t controller::DOWN

6.3.1.7 int32_t controller::fd

6.3.1.8 int8_t controller::HOME

6.3.1.9 int8_t controller::LB

6.3.1.10 int8_t controller::LB2

6.3.1.11 int8_t controller::LEFT

6.3.1.12 struct { ... } controller::LJOY

6.3.1.13 float controller::LT

6.3.1.14 `char* controller::name`

6.3.1.15 `int8_t controller::pressed`

6.3.1.16 `int8_t controller::RB`

6.3.1.17 `int8_t controller::RB2`

6.3.1.18 `int8_t controller::RIGHT`

6.3.1.19 `struct { ... } controller::RJOY`

6.3.1.20 `float controller::RT`

6.3.1.21 `int8_t controller::SELECT`

6.3.1.22 `int8_t controller::START`

6.3.1.23 `int8_t controller::UP`

6.3.1.24 `int8_t controller::X`

6.3.1.25 `float controller::x`

6.3.1.26 `int8_t controller::Y`

6.3.1.27 `float controller::y`

The documentation for this struct was generated from the following file:

- SEVATBR-doc/manual/[controller.h](#)

6.4 HBridgeMotor Class Reference

Public Member Functions

- [HBridgeMotor](#) ()
- void [setdigital](#) (bool d)
- void [write](#) (int v)
- int [attach](#) (int pin1, int pin2)
- void [reset](#) ()

Public Attributes

- short [velocity](#)
- char [pin](#) [2]
- bool [isdigital](#)

6.4.1 Constructor & Destructor Documentation

6.4.1.1 `HBridgeMotor::HBridgeMotor () [inline]`

6.4.2 Member Function Documentation

6.4.2.1 int HBridgeMotor::attach (int *pin1*, int *pin2*) [inline]

6.4.2.2 void HBridgeMotor::reset () [inline]

6.4.2.3 void HBridgeMotor::setdigital (bool *d*) [inline]

6.4.2.4 void HBridgeMotor::write (int *v*) [inline]

6.4.3 Member Data Documentation

6.4.3.1 bool HBridgeMotor::isdigital

6.4.3.2 char HBridgeMotor::pin[2]

6.4.3.3 short HBridgeMotor::velocity

The documentation for this class was generated from the following file:

- SEVATBR-doc/robot/arduino/wheels/[wheels.cpp](#)

6.5 httpLink Struct Reference

```
#include <httpLink.h>
```

Public Attributes

- char * [hostname](#)
- char [ipaddr](#) [128]
- int [socket_fd](#)
- int [connected](#)

6.5.1 Member Data Documentation

6.5.1.1 int httpLink::connected

6.5.1.2 char* httpLink::hostname

6.5.1.3 char httpLink::ipaddr[128]

6.5.1.4 int httpLink::socket_fd

The documentation for this struct was generated from the following file:

- SEVATBR-doc/manual/[httpLink.h](#)

6.6 info Struct Reference

Public Attributes

- [speech_signal_t](#) ss

6.6.1 Member Data Documentation

6.6.1.1 `speech_signal_t` info::ss

The documentation for this struct was generated from the following file:

- SEVATBR-doc/agent/[agent.c](#)

6.7 note Union Reference

Public Attributes

- struct {
 char [low](#)
 char [high](#)
};
- int16_t [val](#)

6.7.1 Member Data Documentation

6.7.1.1 `struct { ... }`

6.7.1.2 `char` note::high

6.7.1.3 `char` note::low

6.7.1.4 `int16_t` note::val

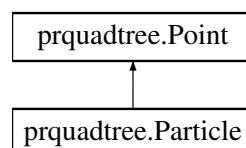
The documentation for this union was generated from the following file:

- SEVATBR-doc/speech/speech-to-text/[lse_stt.c](#)

6.8 prquadtree.Particle Class Reference

Represents particle point.

Inheritance diagram for prquadtree.Particle:



Public Member Functions

- def [__init__](#)
 Constructs a [Particle](#).

Public Attributes

- [x](#)
- [y](#)
- [score](#)

6.8.1 Detailed Description

Represents particle point.

6.8.2 Constructor & Destructor Documentation

6.8.2.1 `def prquadtree.Particle.__init__(self, x, y)`

Constructs a [Particle](#).

Parameters

x	float/int x-position
y	float/int y-position

6.8.3 Member Data Documentation

6.8.3.1 `prquadtree.Particle.score`

6.8.3.2 `prquadtree.Particle.x`

6.8.3.3 `prquadtree.Particle.y`

The documentation for this class was generated from the following file:

- SEVATBR-doc/visual/[prquadtree.py](#)

6.9 particlefilter.ParticleFilter Class Reference

Public Member Functions

- `def __init__`
Constructor.
- `def iterate`
For each blob, it updates the points in the tree increasing the score of those which are within the bounding square of the blob.
- `def score`
Returns the sum of the scores of the points found within this blob by querying the quadtree.
- `def clear_scores`
Resets all scores of blobs This should be used when changing the webcam view.

Public Attributes

- [pr_tree](#)
- [image_box](#)
- [iterations](#)
- [iterations_before_clearing](#)

6.9.1 Constructor & Destructor Documentation

6.9.1.1 `def particlefilter.ParticleFilter.__init__(self, box)`

Constructor.

Parameters

<i>box</i>	Box the box representing the web cam view
------------	---

6.9.2 Member Function Documentation

6.9.2.1 `def particlefilter.ParticleFilter.clear_scores (self)`

Resets all scores of blobs This should be used when changing the webcam view.

6.9.2.2 `def particlefilter.ParticleFilter.iterate (self, blobs)`

For each blob, it updates the points in the tree increasing the score of those which are within the bounding square of the blob.

Parameters

<i>blobs</i>	array An array of blob objects which were matched
--------------	---

6.9.2.3 `def particlefilter.ParticleFilter.score (self, blob)`

Returns the sum of the scores of the points found within this blob by querying the quadtree.

Parameters

<i>blob</i>	Blob A single blob
-------------	--------------------

Returns

int The sum of the scores of the points contained in the passed blob

6.9.3 Member Data Documentation

6.9.3.1 `particlefilter.ParticleFilter.image_box`

6.9.3.2 `particlefilter.ParticleFilter.iterations`

6.9.3.3 `particlefilter.ParticleFilter.iterations_before_clearing`

6.9.3.4 `particlefilter.ParticleFilter.pr_tree`

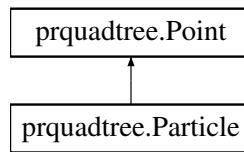
The documentation for this class was generated from the following file:

- SEVATBR-doc/visual/[particlefilter.py](#)

6.10 `prquadtree.Point` Class Reference

Represents an (x,y) coordinate point on a grid.

Inheritance diagram for prquadtree.Point:



Public Member Functions

- `def __init__`
Constructs a coordinate [Point](#).
- `def __str__`
Overwriting the default to string method of the [Point](#) class.
- `def __repr__`
Needed for printing via 'print'.

Public Attributes

- [x](#)
- [y](#)

6.10.1 Detailed Description

Represents an (x,y) coordinate point on a grid.

6.10.2 Constructor & Destructor Documentation

6.10.2.1 `def prquadtree.Point.__init__(self, x, y)`

Constructs a coordinate [Point](#).

Parameters

x	float/int x-position
y	float/int y-position

6.10.3 Member Function Documentation

6.10.3.1 `def prquadtree.Point.__repr__(self)`

Needed for printing via 'print'.

6.10.3.2 `def prquadtree.Point.__str__(self)`

Overwriting the default to string method of the [Point](#) class.

Returns

String representation of [Point](#)

6.10.4 Member Data Documentation

6.10.4.1 `prquadtree.Point.x`

6.10.4.2 `prquadtree.Point.y`

The documentation for this class was generated from the following file:

- SEVATBR-doc/visual/[prquadtree.py](#)

6.11 point2d Struct Reference

```
#include <coord.h>
```

Public Attributes

- double [x](#)
- double [y](#)

6.11.1 Member Data Documentation

6.11.1.1 `double point2d::x`

6.11.1.2 `double point2d::y`

The documentation for this struct was generated from the following file:

- SEVATBR-doc/core/[coord.h](#)

6.12 point3d Struct Reference

```
#include <coord.h>
```

Public Attributes

- double [x](#)
- double [y](#)
- double [z](#)

6.12.1 Member Data Documentation

6.12.1.1 `double point3d::x`

6.12.1.2 `double point3d::y`

6.12.1.3 `double point3d::z`

The documentation for this struct was generated from the following file:

- SEVATBR-doc/core/[coord.h](#)

6.13 pose2d Struct Reference

```
#include <coord.h>
```

Public Attributes

- double [x](#)
- double [y](#)
- double [theta](#)

6.13.1 Member Data Documentation

6.13.1.1 double [pose2d::theta](#)

6.13.1.2 double [pose2d::x](#)

6.13.1.3 double [pose2d::y](#)

The documentation for this struct was generated from the following file:

- SEVATBR-doc/core/[coord.h](#)

6.14 pose3d Struct Reference

```
#include <coord.h>
```

Public Attributes

- double [x](#)
- double [y](#)
- double [z](#)
- double [yaw](#)
- double [pitch](#)
- double [roll](#)

6.14.1 Member Data Documentation

6.14.1.1 double [pose3d::pitch](#)

6.14.1.2 double [pose3d::roll](#)

6.14.1.3 double [pose3d::x](#)

6.14.1.4 double [pose3d::y](#)

6.14.1.5 double [pose3d::yaw](#)

6.14.1.6 double [pose3d::z](#)

The documentation for this struct was generated from the following file:

- SEVATBR-doc/core/[coord.h](#)

6.15 prquadtree.PRQuadTree Class Reference

Class representing a [Point](#) Range Quadtree.

Public Member Functions

- def [__init__](#)
Constructs a PR Quadtree given an initial square.
- def [insert](#)
Inserts a point into the PRQuadtree.
- def [query_range](#)
Returns the points in the provided range.
- def [query_k_nearest](#)
Returns k points closest to the provided point.
- def [print_all_points](#)
Prints all points stored in the PRQuadtree.
- def [__str__](#)
Prints the points of the nw,ne,sw,se blocks of the given [PRQuadTree](#) node.

Static Public Member Functions

- def [size](#)
Static method that determines the size of the given tree.

Public Attributes

- [box](#)
- [points](#)
- [nw](#)
- [ne](#)
- [sw](#)
- [se](#)

Static Public Attributes

- int [QT_NODE_CAPACITY](#) = 20

Private Member Functions

- def [_subdivide](#)
Divides a node into nw,ne,sw,se pieces so that a new point can be inserted.

6.15.1 Detailed Description

Class representing a [Point](#) Range Quadtree.

6.15.2 Constructor & Destructor Documentation

6.15.2.1 def prquadtree.PRQuadTree.__init__(self, box)

Constructs a PR Quadtree given an initial square.

Parameters

<i>box</i>	Box representing initial square
------------	---

6.15.3 Member Function Documentation

6.15.3.1 def prquadtree.PRQuadTree.__str__(self)

Prints the points of the nw,ne,sw,se blocks of the given [PRQuadTree](#) node.

Returns

String A string of points in the blocks Generates string based on the number of points stored in the provided node.

Parameters

<i>loc</i>	PRQuadTree a PRQuadTree node (ie nw,ne,sw,se)
<i>name</i>	String

Returns

String A string with point and name

6.15.3.2 def prquadtree.PRQuadTree._subdivide(self) [private]

Divides a node into nw,ne,sw,se pieces so that a new point can be inserted.

6.15.3.3 def prquadtree.PRQuadTree.insert(self, point)

Inserts a point into the PRQuadtree.

Parameters

<i>point</i>	Point
--------------	-----------------------

Returns

A boolean returning true on success, false on failure.

6.15.3.4 def prquadtree.PRQuadTree.print_all_points(self, root)

Prints all points stored in the PRQuadtree.

Parameters

<i>root</i>	PRQuadTree start point, or the root of the Quadtree
-------------	---

Returns

String a string with coordinates

6.15.3.5 def prquadtree.PRQuadTree.query_k_nearest(self, point, k)

Returns k points closest to the provided point.

Parameters

<i>point</i>	Point a Point from which to search for other points.
<i>k</i>	int number of closest points to return

Returns

array A coordinate distance between the search point and the provided point Internal method used to provide python method with a key (coordinate distance) on which to sort.

Parameters

<i>p</i>	Point
----------	-----------------------

Returns

float

6.15.3.6 `def prquadtree.PRQuadTree.query_range (self, rng)`

Returns the points in the provided range.

Parameters

<i>rng</i>	Box a Box range from which to retrieve points
------------	---

Returns

A list of points within the provided range

6.15.3.7 `def prquadtree.PRQuadTree.size (prtree) [static]`

Static method that determines the size of the given tree.

Keeping an insertion count in the client code would be preferred to this due to heavy recursion.

Parameters

<i>prtree</i>	PRQuadTree
---------------	----------------------------

Returns

int An integer representing the number of points in the given tree.

6.15.4 Member Data Documentation

6.15.4.1 `prquadtree.PRQuadTree.box`6.15.4.2 `prquadtree.PRQuadTree.ne`6.15.4.3 `prquadtree.PRQuadTree.nw`6.15.4.4 `prquadtree.PRQuadTree.points`6.15.4.5 `int prquadtree.PRQuadTree.QT_NODE_CAPACITY = 20 [static]`6.15.4.6 `prquadtree.PRQuadTree.se`

6.15.4.7 prquadtree.PRQuadTree.sw

The documentation for this class was generated from the following file:

- SEVATBR-doc/visual/[prquadtree.py](#)

6.16 rawrec Struct Reference

```
#include <rawrec.h>
```

Public Attributes

- void * [pipeline](#)
- void * [source](#)
- void * [sink](#)
- void * [buffer](#)
- void * [caps](#)
- char * [filesink_loc](#)

6.16.1 Member Data Documentation

6.16.1.1 void* rawrec::buffer

6.16.1.2 void* rawrec::caps

6.16.1.3 char* rawrec::filesink_loc

6.16.1.4 void* rawrec::pipeline

6.16.1.5 void* rawrec::sink

6.16.1.6 void* rawrec::source

The documentation for this struct was generated from the following file:

- SEVATBR-doc/speech/speech-to-text/[rawrec.h](#)

6.17 serial Struct Reference

```
#include <serial.h>
```

Public Attributes

- char * [port](#)
- int [fd](#)
- int8_t [connected](#)
- int [baudrate](#)
- int [parity](#)
- char [buffer](#) [SWBUFMAX]
- char [readbuf](#) [SWREADMAX]
- int8_t [readAvailable](#)

6.17.1 Member Data Documentation

6.17.1.1 int serial::baudrate

6.17.1.2 char serial::buffer[SWBUFMAX]

6.17.1.3 int8_t serial::connected

6.17.1.4 int serial::fd

6.17.1.5 int serial::parity

6.17.1.6 char* serial::port

6.17.1.7 int8_t serial::readAvailable

6.17.1.8 char serial::readbuf[SWREADMAX]

The documentation for this struct was generated from the following file:

- SEVATBR-doc/robot/serial/[serial.h](#)

6.18 speech_signal Struct Reference

```
#include <speech_signal.h>
```

Public Attributes

- unsigned char [none](#)
- unsigned char [go](#)
- unsigned char [stop](#)
- unsigned char [fetch](#)
- unsigned char [ret](#)

6.18.1 Member Data Documentation

6.18.1.1 unsigned char speech_signal::fetch

6.18.1.2 unsigned char speech_signal::go

6.18.1.3 unsigned char speech_signal::none

6.18.1.4 unsigned char speech_signal::ret

6.18.1.5 unsigned char speech_signal::stop

The documentation for this struct was generated from the following file:

- SEVATBR-doc/speech/speech-to-text/[speech_signal.h](#)

6.19 stt Struct Reference

```
#include <lse_stt.h>
```

Public Attributes

- char [nothing](#)
- ps_decoder_t * [ps](#)
- cmd_in_t * [config](#)

6.19.1 Member Data Documentation

6.19.1.1 `cmd_in_t*` `stt::config`

6.19.1.2 `char` `stt::nothing`

6.19.1.3 `ps_decoder_t*` `stt::ps`

The documentation for this struct was generated from the following files:

- SEVATBR-doc/speech/speech-to-text/[lse_stt.h](#)
- SEVATBR-doc/speech/speech-to-text/[stt.h](#)

6.20 tbr Struct Reference

```
#include <tbr.h>
```

Public Attributes

- `serial_t` * [connections](#)
- int * [ids](#)
- `int8_t` [connected](#)
- char ** [possible_ports](#)
- int [num_possible](#)
- int [left](#)
- int [right](#)
- int [arm](#)
- int [claw](#)
- int [prev_left](#)
- int [prev_right](#)
- int [prev_arm](#)
- int [prev_claw](#)

6.20.1 Member Data Documentation

6.20.1.1 `int` `tbr::arm`

6.20.1.2 `int` `tbr::claw`

6.20.1.3 `int8_t` `tbr::connected`

6.20.1.4 `serial_t*` `tbr::connections`

6.20.1.5 `int*` `tbr::ids`

6.20.1.6 `int` `tbr::left`

6.20.1.7 `int tbr::num_possible`

6.20.1.8 `char** tbr::possible_ports`

6.20.1.9 `int tbr::prev_arm`

6.20.1.10 `int tbr::prev_claw`

6.20.1.11 `int tbr::prev_left`

6.20.1.12 `int tbr::prev_right`

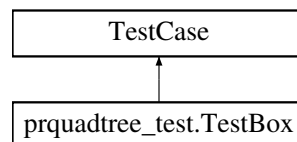
6.20.1.13 `int tbr::right`

The documentation for this struct was generated from the following file:

- SEVATBR-doc/robot/tbr/[tbr.h](#)

6.21 prquadtree_test.TestBox Class Reference

Inheritance diagram for prquadtree_test.TestBox:



Public Member Functions

- def [test_box_insert](#)
- def [test_box_contains](#)

6.21.1 Member Function Documentation

6.21.1.1 `def prquadtree_test.TestBox.test_box_contains (self)`

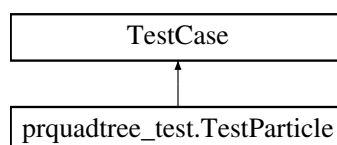
6.21.1.2 `def prquadtree_test.TestBox.test_box_insert (self)`

The documentation for this class was generated from the following file:

- SEVATBR-doc/visual/[prquadtree_test.py](#)

6.22 prquadtree_test.TestParticle Class Reference

Inheritance diagram for prquadtree_test.TestParticle:



Public Member Functions

- def [test_particle_insert](#)

6.22.1 Member Function Documentation

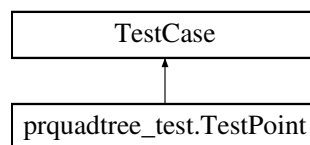
6.22.1.1 def prquadtree_test.TestParticle.test_particle_insert (*self*)

The documentation for this class was generated from the following file:

- SEVATBR-doc/visual/[prquadtree_test.py](#)

6.23 prquadtree_test.TestPoint Class Reference

Inheritance diagram for prquadtree_test.TestPoint:



Public Member Functions

- def [test_point_insert](#)

6.23.1 Member Function Documentation

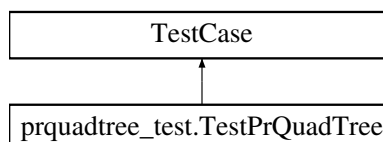
6.23.1.1 def prquadtree_test.TestPoint.test_point_insert (*self*)

The documentation for this class was generated from the following file:

- SEVATBR-doc/visual/[prquadtree_test.py](#)

6.24 prquadtree_test.TestPrQuadTree Class Reference

Inheritance diagram for prquadtree_test.TestPrQuadTree:



Public Member Functions

- def [test_insert](#)
- def [test_nearby](#)

6.24.1 Member Function Documentation

6.24.1.1 `def prquadtree_test.TestPrQuadTree.test_insert (self)`

6.24.1.2 `def prquadtree_test.TestPrQuadTree.test_nearby (self)`

The documentation for this class was generated from the following file:

- SEVATBR-doc/visual/[prquadtree_test.py](#)

Chapter 7

File Documentation

7.1 SEVATBR-doc/agent/agent.c File Reference

```
#include <pthread.h>
#include "speech.h"
#include "visual.h"
#include "coord.h"
#include "parietal.h"
```

Classes

- struct [info](#)
- struct [_ctrl](#)

Typedefs

- typedef struct [info](#) [info_t](#)

Functions

- void [wakeup](#) ()
- void * [conscious_thought](#) (void *information)
- void [gotosleep](#) ()

Variables

- static pthread_t [consciousness](#)
- static int [end_consciousness](#)
- struct [_ctrl](#) [ctrl](#)

7.1.1 Typedef Documentation

7.1.1.1 typedef struct info info_t

7.1.2 Function Documentation

7.1.2.1 void* concious_thought (void * *information*)

Concious thought thread

Parameters

<i>information</i>	info for the thought
--------------------	----------------------

Returns

NULL

7.1.2.2 void gotosleep ()

End threads, go back to sleep

7.1.2.3 void wakeup ()

Wakeup from being asleep, start concious threads

7.1.3 Variable Documentation

7.1.3.1 pthread_t conciousness [static]

7.1.3.2 struct _ctrl ctrl

7.1.3.3 int end_conciousness [static]

7.2 SEVATBR-doc/agent/agent.h File Reference

```
#include "coord.h"
```

Macros

- #define `AGENT_SIMPLE` 0x0001

Functions

- int `agent_create` (int type)
- void `agent_enable` (void)
- void `agent_disable` (void)
- void `agent_destroy` (void)
- void `agent_get_poses` (pose3d_t *base, pose3d_t *arm)

7.2.1 Macro Definition Documentation

7.2.1.1 #define AGENT_SIMPLE 0x0001

7.2.2 Function Documentation

7.2.2.1 int agent_create (int type)

Create the agent

Parameters

<i>type</i>	the type to start the agent with
-------------	----------------------------------

7.2.2.2 void agent_destroy (void)

Destroy the agent

7.2.2.3 void agent_disable (void)

Disable the agent

7.2.2.4 void agent_enable (void)

Enable the agent

7.2.2.5 void agent_get_poses (pose3d_t * base, pose3d_t * arm)

Get the poses from the agent

Parameters

<i>base</i>	the base pose to set
<i>arm</i>	the arm pose to set

7.3 SEVATBR-doc/agent/simple_agent.cpp File Reference

```
#include <sys/time.h>
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include "agent.h"
#include "tbd.h"
```

Functions

- int [agent_create](#) (int type)
- void [agent_enable](#) (void)
- void [agent_disable](#) (void)
- void [agent_destroy](#) (void)
- void [agent_get_poses](#) (pose3d_t *base, pose3d_t *arm)

Variables

- static int [agent_enabled](#)
- static int [view](#) [2]
- static struct timeval [last_signal](#)

7.3.1 Function Documentation

7.3.1.1 `int agent_create (int type)`

Create the agent

Parameters

<i>type</i>	the type to start the agent with
-------------	----------------------------------

7.3.1.2 void agent_destroy (void)

Destroy the agent

7.3.1.3 void agent_disable (void)

Disable the agent

7.3.1.4 void agent_enable (void)

Enable the agent

7.3.1.5 void agent_get_poses (pose3d_t * base, pose3d_t * arm)

Get the poses from the agent

Parameters

<i>base</i>	the base pose to set
<i>arm</i>	the arm pose to set

7.3.2 Variable Documentation

7.3.2.1 int agent_enabled [static]

7.3.2.2 struct timeval last_signal [static]

7.3.2.3 int view[2] [static]

7.4 SEVATBR-doc/agent/test.c File Reference

```
#include <stdio.h>
#include <signal.h>
#include <sys/time.h>
#include "tbd.h"
```

Functions

- void [stopprog](#) (int signum)
- int [main](#) (int argc, char *argv[])

Variables

- static int [exit_signal](#)

7.4.1 Function Documentation

7.4.1.1 `int main (int argc, char * argv[])`

7.4.1.2 `void stopprog (int signum)`

7.4.2 Variable Documentation

7.4.2.1 `int exit_signal [static]`

7.5 SEVATBR-doc/manual/test.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <signal.h>
#include "manual.h"
```

Functions

- void `stop` (int *signo*)
- int `main` ()

Variables

- static int `stopsig`

7.5.1 Function Documentation

7.5.1.1 `int main ()`

7.5.1.2 `void stop (int signo)`

7.5.2 Variable Documentation

7.5.2.1 `int stopsig [static]`

7.6 SEVATBR-doc/robot/tbr/test.c File Reference

```
#include "tbr.h"
```

Functions

- int `main` ()

7.6.1 Function Documentation

7.6.1.1 `int main ()`

7.7 SEVATBR-doc/speech/speech-to-text/test.c File Reference

```
#include <stdio.h>
#include <signal.h>
#include "speech_signal.h"
```

Functions

- void [stop](#) (int param)
- int [main](#) (int argc, char **argv)

Variables

- static unsigned char [stopsig](#)

7.7.1 Function Documentation

7.7.1.1 int [main](#) (int *argc*, char ** *argv*)

7.7.1.2 void [stop](#) (int *param*)

7.7.2 Variable Documentation

7.7.2.1 unsigned char [stopsig](#) [static]

7.8 SEVATBR-doc/core/coord.h File Reference

Classes

- struct [pose3d](#)
- struct [pose2d](#)
- struct [point3d](#)
- struct [point2d](#)

Typedefs

- typedef struct [pose3d](#) [pose3d_t](#)
- typedef struct [pose2d](#) [pose2d_t](#)
- typedef struct [point3d](#) [point3d_t](#)
- typedef struct [point2d](#) [point2d_t](#)
- typedef [point2d_t](#) [point_t](#)
- typedef [pose2d_t](#) [pose_t](#)

7.8.1 Typedef Documentation

7.8.1.1 typedef struct [point2d](#) [point2d_t](#)

7.8.1.2 typedef struct [point3d](#) [point3d_t](#)

7.8.1.3 typedef [point2d_t](#) [point_t](#)

7.8.1.4 `typedef struct pose2d pose2d_t`

7.8.1.5 `typedef struct pose3d pose3d_t`

7.8.1.6 `typedef pose2d_t pose_t`

7.9 SEVATBR-doc/core/core.c File Reference

```
#include <signal.h>
#include <string.h>
#include <stdio.h>
#include "robot.h"
#include "manual.h"
#include "coord.h"
```

Functions

- void [stop_program](#) (int signum)
- int [main](#) (int argc, char *argv[])

Variables

- static int [stop_signal](#)

7.9.1 Function Documentation

7.9.1.1 `int main (int argc, char * argv[])`

This is the starting program for the robot

Parameters

<i>argc</i>	standard exec argument number
<i>argv</i>	standard exec argument list

Returns

0 on success, -1 otherwise

7.9.1.2 `void stop_program (int signum)`

Signal handler to stop the program

Parameters

<i>signum</i>	the signal number (kernel dependent)
---------------	--------------------------------------

7.9.2 Variable Documentation

7.9.2.1 `int stop_signal` [`static`]

7.10 SEVATBR-doc/core/core.cpp File Reference

```
#include <signal.h>
#include <string.h>
#include <stdio.h>
#include "robot.h"
#include "manual.h"
#include "coord.h"
```

Functions

- void [stop_program](#) (int signum)
- int [main](#) (int argc, char *argv[])

Variables

- static int [stop_signal](#)

7.10.1 Function Documentation

7.10.1.1 int main (int *argc*, char * *argv*[])

This is the starting program for the robot

Parameters

<i>argc</i>	standard exec argument number
<i>argv</i>	standard exec argument list

Returns

0 on success, -1 otherwise

7.10.1.2 void stop_program (int *signum*)

Signal handler to stop the program

Parameters

<i>signum</i>	the signal number (kernel dependent)
---------------	--------------------------------------

7.10.2 Variable Documentation

7.10.2.1 int stop_signal [static]

7.11 SEVATBR-doc/logger/logger.c File Reference

```
#include "logger.h"
```

Functions

- void [logger_init](#) (char *filename)
- void [logger_print](#) (char *msg)
- char * [logger_scan](#) (void)
- void [logger_destroy](#) (void)

Variables

- static FILE * [logfile](#)

7.11.1 Function Documentation

7.11.1.1 void logger_destroy (void)

Stop the logger

7.11.1.2 void logger_init (char * fname)

Start the logger

Parameters

<i>fname</i>	name of the input file
--------------	------------------------

7.11.1.3 void logger_print (char * msg)

Print something to the logger

Parameters

<i>msg</i>	the message to print to the file
------------	----------------------------------

7.11.1.4 char * logger_scan (void)

Get the current logger file

Returns

the entire message on success, NULL otherwise

7.11.2 Variable Documentation

7.11.2.1 FILE* logfile [static]

7.12 SEVATBR-doc/logger/logger.h File Reference

Functions

- void [log_init](#) (void)
- void [log_print](#) (char *msg)
- char * [log_scan](#) (void)
- void [log_destroy](#) (void)

7.12.1 Function Documentation

7.12.1.1 void log_destroy (void)

7.12.1.2 void log_init (void)

7.12.1.3 void log_print (char * msg)

7.12.1.4 char* log_scan (void)

7.13 SEVATBR-doc/manual/controller.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <dirent.h>
#include <linux/joystick.h>
#include <string.h>
#include <unistd.h>
#include <fcntl.h>
#include "controller.h"
```

Macros

- #define INPUT_DIR "/dev/input/"
- #define JS_PREFIX "js"
- #define MAX_16BIT 0x7FFF

Functions

- int controller_update (controller_t *ctrl)
- void controller_connect (controller_t *ctrl)
- void controller_disconnect (controller_t *ctrl)

7.13.1 Macro Definition Documentation

7.13.1.1 #define INPUT_DIR "/dev/input/"

7.13.1.2 #define JS_PREFIX "js"

7.13.1.3 #define MAX_16BIT 0x7FFF

7.13.2 Function Documentation

7.13.2.1 void controller_connect (controller_t * ctrl)

Connect to a joystick device.

Parameters

<i>controller</i>	A pointer to a controller struct.
-------------------	-----------------------------------

7.13.2.2 void controller_disconnect (controller_t * ctrl)

Disconnect from a joystick device.

Parameters

<i>controller</i>	A pointer to a controller struct.
-------------------	-----------------------------------

7.13.2.3 int controller_update (controller_t * ctrl)

Hidden. Update a joystick device asynchronously.

Parameters

<i>controller_arg</i>	A (void *) pointer to a controller struct.
-----------------------	--

Returns

0 on success, else -1

7.14 SEVATBR-doc/manual/controller.h File Reference

```
#include <stdint.h>
#include <pthread.h>
```

Classes

- struct [controller](#)

Typedefs

- typedef struct [controller](#) [controller_t](#)

Functions

- void [controller_connect](#) ([controller_t](#) *ctrl)
- void [controller_disconnect](#) ([controller_t](#) *ctrl)

7.14.1 Typedef Documentation

7.14.1.1 typedef struct controller controller_t

7.14.2 Function Documentation

7.14.2.1 void controller_connect (controller_t * ctrl)

Connect to a joystick device.

Parameters

<i>controller</i>	A pointer to a controller struct.
-------------------	-----------------------------------

7.14.2.2 void controller_disconnect (controller_t * ctrl)

Disconnect from a joystick device.

Parameters

<i>controller</i>	A pointer to a controller struct.
-------------------	-----------------------------------

7.15 SEVATBR-doc/manual/httplink.c File Reference

```
#include <stdio.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <stdlib.h>
#include <netdb.h>
#include <string.h>
#include <unistd.h>
#include "httplink.h"
```

Functions

- int [httplink_connect](#) ([httplink_t](#) *connection, char *hostname)
- int [httplink_send](#) ([httplink_t](#) *connection, char *addr, char *type, char *data)
- char * [httplink_recv](#) ([httplink_t](#) *connection)
- int [httplink_disconnect](#) ([httplink_t](#) *connection)

Variables

- static char [msgbuf](#) [1024]
- static char [response](#) [1024]

7.15.1 Function Documentation

7.15.1.1 int httplink_connect ([httplink_t](#) * *connection*, char * *hostname*)

Connect to the main server for our robot's manual interface.

Parameters

<i>connection</i>	the connection information for the server
<i>hostname</i>	the hostname of the server

Returns

0 on success, -1 otherwise

7.15.1.2 int httplink_disconnect ([httplink_t](#) * *connection*)

Disconnect the connection

Parameters

<i>connection</i>	the connection information for the server
-------------------	---

7.15.1.3 char* httplink_recv ([httplink_t](#) * *connection*)

Try and receive a response from the main server

Parameters

<i>connection</i>	the connection information for the server
-------------------	---

Returns

n bytes received, -1 otherwise

7.15.1.4 `int httplink_send (httplink_t * connection, char * addr, char * type, char * data)`

Send a request to the main server

Parameters

<i>connection</i>	the connection information for the server
<i>addr</i>	the addr to send the request to, "/" for main page
<i>type</i>	either "GET" or "POST" or "get" or "post"
<i>data</i>	the data to send over (only for httplink_POST)

Returns

n bytes sent over, -1 otherwise

7.15.2 Variable Documentation

7.15.2.1 `char msgbuf[1024] [static]`

7.15.2.2 `char response[1024] [static]`

7.16 SEVATBR-doc/manual/httplink.h File Reference

Classes

- struct [httplink](#)

Typedefs

- typedef struct [httplink](#) [httplink_t](#)

Functions

- int [httplink_connect](#) ([httplink_t](#) **connection*, char **hostname*)
- int [httplink_send](#) ([httplink_t](#) **connection*, char **addr*, char **type*, char **data*)
- char * [httplink_recv](#) ([httplink_t](#) **connection*)
- int [httplink_disconnect](#) ([httplink_t](#) **connection*)

7.16.1 Typedef Documentation

7.16.1.1 `typedef struct httplink httplink_t`

7.16.2 Function Documentation

7.16.2.1 `int httplink_connect (httplink_t * connection, char * hostname)`

Connect to the main server for our robot's manual interface.

Parameters

<i>connection</i>	the connection information for the server
<i>hostname</i>	the hostname of the server

Returns

0 on success, -1 otherwise

7.16.2.2 int httplink_disconnect (httplink_t * connection)

Disconnect the connection

Parameters

<i>connection</i>	the connection information for the server
-------------------	---

7.16.2.3 char* httplink_recv (httplink_t * connection)

Try and receive a response from the main server

Parameters

<i>connection</i>	the connection information for the server
-------------------	---

Returns

n bytes received, -1 otherwise

7.16.2.4 int httplink_send (httplink_t * connection, char * addr, char * type, char * data)

Send a request to the main server

Parameters

<i>connection</i>	the connection information for the server
<i>addr</i>	the addr to send the request to, "/" for main page
<i>type</i>	either "GET" or "POST" or "get" or "post"
<i>data</i>	the data to send over (only for httplink_POST)

Returns

n bytes sent over, -1 otherwise

7.17 SEVATBR-doc/manual/manual.c File Reference

```
#include <stdio.h>
#include <signal.h>
#include <string.h>
#include <sys/time.h>
#include <stdlib.h>
#include "httplink.h"
#include "controller.h"
#include "manual.h"
```

Macros

- `#define HZ 10`

Functions

- static void `server_update` (void)
- static void `raise_server_request` (int signum)
- static void `controller_update` (void)
- int `manual_connect` (int id)
- void `manual_enable` (void)
- void `manual_disable` (void)
- int `manual_disconnect` (void)
- int `manual_new_data` (void)
- int `isOverriden` (void)
- void `manual_get_poses` (pose3d_t *b, pose3d_t *a)

Variables

- static int `input_id`
- static `httplink_t` `server`
- static `controller_t` `ctrl`
- static `pose3d_t` `base`
- static `pose3d_t` `arm`
- static int `new_join`
- static struct timeval `last_signal`
- static int `manual_en`
- static int `mnl_override`

7.17.1 Macro Definition Documentation

7.17.1.1 `#define HZ 10`

7.17.2 Function Documentation

7.17.2.1 `void controller_update (void) [static]`

Private to set the base and arm using information from the controller

7.17.2.2 `int isOverriden (void)`

User override

Returns

1 if overridden, else 0

7.17.2.3 `int manual_connect (int id)`

Connect to the manual connection - do not enable just yet.

Parameters

<i>id</i>	the id of the manual connection to connect to
-----------	---

Returns

0 on success, -1 otherwise

7.17.2.4 void manual_disable (void)

Disable manual mode

7.17.2.5 int manual_disconnect (void)

Disconnect from the manual connection

Returns

0, else -1 on error

7.17.2.6 void manual_enable (void)

Enable manual mode

7.17.2.7 void manual_get_poses (pose3d_t * b, pose3d_t * a)

Get the poses

Parameters

<i>the</i>	structs needed to hold the poses
------------	----------------------------------

7.17.2.8 int manual_new_data (void)

Get the status of the join

Returns

1 if a new join exists, else 0

7.17.2.9 static void raise_server_request (int *signum*) [static]

Private method to handle server requesting

Parameters

<i>signum</i>	the id for the signal
---------------	-----------------------

7.17.2.10 static void server_update (void) [static]

Private method to get the information sent over from the server and set the robot with this information

7.17.3 Variable Documentation

7.17.3.1 `pose3d_t arm` `[static]`

7.17.3.2 `pose3d_t base` `[static]`

7.17.3.3 `controller_t ctrl` `[static]`

7.17.3.4 `int input_id` `[static]`

7.17.3.5 `struct timeval last_signal` `[static]`

7.17.3.6 `int manual_en` `[static]`

7.17.3.7 `int mnl_override` `[static]`

7.17.3.8 `int new_join` `[static]`

7.17.3.9 `httplink_t server` `[static]`

7.18 SEVATBR-doc/manual/manual.h File Reference

```
#include "coord.h"
```

Macros

- `#define MNL_SRVR 0x0001`
- `#define MNL_CTRL 0x0002`

Functions

- `int manual_connect (int type)`
- `void manual_enable (void)`
- `void manual_disable (void)`
- `int manual_disconnect (void)`
- `int manual_new_data (void)`
- `void manual_get_poses (pose3d_t *base, pose3d_t *arm)`
- `int isOverriden (void)`

7.18.1 Macro Definition Documentation

7.18.1.1 `#define MNL_CTRL 0x0002`

7.18.1.2 `#define MNL_SRVR 0x0001`

7.18.2 Function Documentation

7.18.2.1 `int isOverriden (void)`

User override

Returns

1 if overridden, else 0

7.18.2.2 int manual_connect (int *id*)

Connect to the manual connection - do not enable just yet.

Parameters

<i>id</i>	the id of the manual connection to connect to
-----------	---

Returns

0 on success, -1 otherwise

7.18.2.3 void manual_disable (void)

Disable manual mode

7.18.2.4 int manual_disconnect (void)

Disconnect from the manual connection

Returns

0, else -1 on error

7.18.2.5 void manual_enable (void)

Enable manual mode

7.18.2.6 void manual_get_poses (pose3d_t * *b*, pose3d_t * *a*)

Get the poses

Parameters

<i>the</i>	structs needed to hold the poses
------------	----------------------------------

7.18.2.7 int manual_new_data (void)

Get the status of the join

Returns

1 if a new join exists, else 0

7.19 SEVATBR-doc/manual/rtsp/rtsplink.c File Reference

```
#include <gst/gst.h>
```

7.20 SEVATBR-doc/robot/arduino/arm/arm.cpp File Reference

```
#include <Servo.h>
#include <string.h>
```

Macros

- `#define DEV_ID` 2
- `#define ARM_R1` 3
- `#define ARM_L1` 5
- `#define ARM_R2` 6
- `#define ARM_L2` 9
- `#define ARM_R3` 10
- `#define ARM_L3` 11

Functions

- `int limit` (int *x*, int *a*, int *b*)
- `void setarm` (int *vel*)
- `void stoparm` ()
- `void raisearm` ()
- `void lowerarm` ()
- `void setup` ()
- `void loop` ()

Variables

- `Servo arm_l` [3]
- `Servo arm_r` [3]
- unsigned long `time`
- char `msg` [64]

7.20.1 Macro Definition Documentation

7.20.1.1 `#define ARM_L1` 5

7.20.1.2 `#define ARM_L2` 9

7.20.1.3 `#define ARM_L3` 11

7.20.1.4 `#define ARM_R1` 3

7.20.1.5 `#define ARM_R2` 6

7.20.1.6 `#define ARM_R3` 10

7.20.1.7 `#define DEV_ID` 2

7.20.2 Function Documentation

7.20.2.1 `int limit` (int *x*, int *a*, int *b*)

7.20.2.2 void loop ()

7.20.2.3 void lowerarm ()

7.20.2.4 void raisearm ()

7.20.2.5 void setarm (int vel)

7.20.2.6 void setup ()

7.20.2.7 void stoparm ()

7.20.3 Variable Documentation

7.20.3.1 Servo arm_l[3]

7.20.3.2 Servo arm_r[3]

7.20.3.3 char msg[64]

7.20.3.4 unsigned long time

7.21 SEVATBR-doc/robot/arduino/claw/claw.cpp File Reference

```
#include <Servo.h>
#include <string.h>
```

Macros

- #define DEV_ID 3
- #define CLAW_L 10
- #define BOT_L 2
- #define TOP_L 3
- #define CLAW_R 9
- #define BOT_R 0
- #define TOP_R 1

Functions

- int limit (int x, int a, int b)
- void setclaw (int vel)
- void stopclaw ()
- void openclaw ()
- void closeclaw ()
- void setup ()
- void loop ()

Variables

- Servo claw_l
- Servo claw_r
- unsigned long time
- char msg [64]

7.21.1 Macro Definition Documentation

7.21.1.1 `#define BOT_L 2`

7.21.1.2 `#define BOT_R 0`

7.21.1.3 `#define CLAW_L 10`

7.21.1.4 `#define CLAW_R 9`

7.21.1.5 `#define DEV_ID 3`

7.21.1.6 `#define TOP_L 3`

7.21.1.7 `#define TOP_R 1`

7.21.2 Function Documentation

7.21.2.1 `void closeclaw ()`

7.21.2.2 `int limit (int x, int a, int b)`

7.21.2.3 `void loop ()`

7.21.2.4 `void openclaw ()`

7.21.2.5 `void setclaw (int vel)`

7.21.2.6 `void setup ()`

7.21.2.7 `void stopclaw ()`

7.21.3 Variable Documentation

7.21.3.1 `Servo claw_l`

7.21.3.2 `Servo claw_r`

7.21.3.3 `char msg[64]`

7.21.3.4 `unsigned long time`

7.22 SEVATBR-doc/robot/arduino/wheels/wheels.cpp File Reference

```
#include <string.h>
```

Classes

- class [HBridgeMotor](#)

Macros

- `#define` [DEV_ID](#) 1

- `#define WHEEL_RT1 11`
- `#define WHEEL_RT2 10`
- `#define WHEEL_RM1 9`
- `#define WHEEL_RM2 8`
- `#define WHEEL_RB1 7`
- `#define WHEEL_RB2 6`
- `#define WHEEL_LT1 A0`
- `#define WHEEL_LT2 A1`
- `#define WHEEL_LM1 A2`
- `#define WHEEL_LM2 A3`
- `#define WHEEL_LB1 A4`
- `#define WHEEL_LB2 A5`

Functions

- `int limit (int x, int a, int b)`
- `void setwheels (int left, int right)`
- `void stopwheels ()`
- `void turnleft ()`
- `void turnright ()`
- `void forward ()`
- `void backward ()`
- `void setup ()`
- `void loop ()`

Variables

- `HBridgeMotor wheel_l [3]`
- `HBridgeMotor wheel_r [3]`
- `unsigned long time`
- `char msg [64]`

7.22.1 Macro Definition Documentation

7.22.1.1 `#define DEV_ID 1`

7.22.1.2 `#define WHEEL_LB1 A4`

7.22.1.3 `#define WHEEL_LB2 A5`

7.22.1.4 `#define WHEEL_LM1 A2`

7.22.1.5 `#define WHEEL_LM2 A3`

7.22.1.6 `#define WHEEL_LT1 A0`

7.22.1.7 `#define WHEEL_LT2 A1`

7.22.1.8 `#define WHEEL_RB1 7`

7.22.1.9 `#define WHEEL_RB2 6`

7.22.1.10 `#define WHEEL_RM1 9`

7.22.1.11 `#define WHEEL_RM2 8`

7.22.1.12 `#define WHEEL_RT1 11`

7.22.1.13 `#define WHEEL_RT2 10`

7.22.2 Function Documentation

7.22.2.1 `void backward ()`

7.22.2.2 `void forward ()`

7.22.2.3 `int limit (int x, int a, int b)`

7.22.2.4 `void loop ()`

7.22.2.5 `void setup ()`

7.22.2.6 `void setwheels (int left, int right)`

7.22.2.7 `void stopwheels ()`

7.22.2.8 `void turnleft ()`

7.22.2.9 `void turnright ()`

7.22.3 Variable Documentation

7.22.3.1 `char msg[64]`

7.22.3.2 `unsigned long time`

7.22.3.3 `HBridgeMotor wheel_l[3]`

7.22.3.4 `HBridgeMotor wheel_r[3]`

7.23 SEVATBR-doc/robot/robot.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "robot.h"
#include "tbr.h"
```

Functions

- int `robot_set` (uint32_t robotid)
- int `robot_unset` (void)
- int `robot_move` (pose3d_t *base, pose3d_t *arm)

Variables

- static void * `robot`

- static uint32_t [currid](#)

7.23.1 Function Documentation

7.23.1.1 int robot_move (pose3d_t * base, pose3d_t * arm)

Move the robot by some velocity or to some position

Parameters

<i>base</i>	either a direction or velocity (base)
<i>arm</i>	either a direction or velocity (arm)

Returns

0 on success, -1 otherwise

7.23.1.2 int robot_set (uint32_t robotid)

Initialize the robot to some id which specifies the device

Parameters

<i>robotid</i>	the id of the device
----------------	----------------------

Returns

0 on success, -1 otherwise

7.23.1.3 int robot_unset (void)

Remove the robot from the known space

Returns

0 on success, -1 otherwise

7.23.2 Variable Documentation

7.23.2.1 uint32_t currid [static]

7.23.2.2 void* robot [static]

7.24 SEVATBR-doc/robot/robot.h File Reference

```
#include <stdint.h>
#include "coord.h"
```

Macros

- #define [NO_ROBOT](#) 0x00000000
- #define [STANDARD_OUT](#) 0x00000001
- #define [TENNIS_BALL_ROBOT](#) 0x00000002
- #define [TACHIKOMA](#) 0x00000003

Functions

- int [robot_set](#) (uint32_t robotid)
- int [robot_unset](#) (void)
- int [robot_move](#) (pose3d_t *base, pose3d_t *arm)

7.24.1 Macro Definition Documentation

7.24.1.1 `#define NO_ROBOT 0x00000000`

7.24.1.2 `#define STANDARD_OUT 0x00000001`

7.24.1.3 `#define TACHIKOMA 0x00000003`

7.24.1.4 `#define TENNIS_BALL_ROBOT 0x00000002`

7.24.2 Function Documentation

7.24.2.1 `int robot_move (pose3d_t * base, pose3d_t * arm)`

Move the robot by some velocity or to some position

Parameters

<i>base</i>	either a direction or velocity (base)
<i>arm</i>	either a direction or velocity (arm)

Returns

0 on success, -1 otherwise

7.24.2.2 `int robot_set (uint32_t robotid)`

Initialize the robot to some id which specifies the device

Parameters

<i>robotid</i>	the id of the device
----------------	----------------------

Returns

0 on success, -1 otherwise

7.24.2.3 `int robot_unset (void)`

Remove the robot from the known space

Returns

0 on success, -1 otherwise

7.25 SEVATBR-doc/robot/serial/serial.c File Reference

```
#include <termios.h>
#include <sys/ioctl.h>
#include <sys/types.h>
#include <sys/wait.h>
#include <unistd.h>
#include <stdlib.h>
#include <string.h>
#include <fcntl.h>
#include <dirent.h>
#include <stdio.h>
#include "serial.h"
```

Macros

- `#define INPUT_DIR "/dev/"`

Functions

- static int `_serial_setattr (serial_t *connection)`
- static void `_serial_update (serial_t *connection)`
- int `serial_connect (serial_t *connection, char *port, int baudrate)`
- char * `serial_read (serial_t *connection)`
- void `serial_write (serial_t *connection, char *message)`
- void `serial_disconnect (serial_t *connection)`

Variables

- static char const * `PREFIXES` [3]
- static char `tempbuf` [SWREADMAX]

7.25.1 Macro Definition Documentation

7.25.1.1 `#define INPUT_DIR "/dev/"`

7.25.2 Function Documentation

7.25.2.1 static int `_serial_setattr (serial_t * connection)` [static]

Helper method to set the attributes of a serial connection, particularly for the arduino or similar device.

Parameters

<i>connection</i>	the serial port to connect to
-------------------	-------------------------------

Returns

0 on success, -1 on failure

7.25.2.2 static void `_serial_update (serial_t * connection)` [static]

Method to update the readbuf of the serial communication, as well as the connection itself.

Parameters

<i>connection</i>	the serial struct
-------------------	-------------------

Note

the packets will be read in the following format: data
however, the
will be cut off

7.25.2.3 `int serial_connect (serial_t * connection, char * port, int baudrate)`

Connect to a serial device.

Parameters

<i>connection</i>	a pointer to the serial struct
<i>port</i>	a portname; if NULL, will open a random port
<i>baudrate</i>	the bits per second of information to transmit/receive

Returns

0 on success, -1 on failure

7.25.2.4 `void serial_disconnect (serial_t * connection)`

Disconnect from the USB Serial port.

Parameters

<i>connection</i>	A pointer to the serial struct.
-------------------	---------------------------------

7.25.2.5 `char* serial_read (serial_t * connection)`

Read a string from the serial communication link.

Parameters

<i>connection</i>	the serial connection to read a message from
-------------------	--

Returns

the readbuf if a message exists, else NULL

7.25.2.6 `void serial_write (serial_t * connection, char * message)`

Write a message to the serial communication link.

Parameters

<i>connection</i>	the serial communication link to write to
<i>message</i>	the message to send over to the other side

Note

be sure the message has a '
' character

7.25.3 Variable Documentation

7.25.3.1 `char const* PREFIXES[3]` `[static]`

Initial value:

```
= {  
    "ttyACM",  
  
    NULL  
}
```

7.25.3.2 `char tempbuf[SWREADMAX]` `[static]`

7.26 SEVATBR-doc/robot/serial/serial.h File Reference

```
#include <stdint.h>
```

Classes

- struct [serial](#)

Macros

- #define [SWBUFMAX](#) 64
- #define [SWREADMAX](#) 32
- #define [SWWRITEMAX](#) 32

Typedefs

- typedef struct [serial](#) [serial_t](#)

Functions

- int [serial_connect](#) ([serial_t](#) *connection, char *port, int baudrate)
- char * [serial_read](#) ([serial_t](#) *connection)
- void [serial_write](#) ([serial_t](#) *connection, char *message)
- void [serial_disconnect](#) ([serial_t](#) *connection)

7.26.1 Macro Definition Documentation

7.26.1.1 #define [SWBUFMAX](#) 64

7.26.1.2 #define [SWREADMAX](#) 32

7.26.1.3 #define [SWWRITEMAX](#) 32

7.26.2 Typedef Documentation

7.26.2.1 typedef struct [serial](#) [serial_t](#)

7.26.3 Function Documentation

7.26.3.1 `int serial_connect (serial_t * connection, char * port, int baudrate)`

Connect to a serial device.

Parameters

<i>connection</i>	a pointer to the serial struct
<i>port</i>	a portname; if NULL, will open a random port
<i>baudrate</i>	the bits per second of information to transmit/receive

Returns

0 on success, -1 on failure

7.26.3.2 void serial_disconnect (serial_t * connection)

Disconnect from the USB Serial port.

Parameters

<i>connection</i>	A pointer to the serial struct.
-------------------	---------------------------------

7.26.3.3 char* serial_read (serial_t * connection)

Read a string from the serial communication link.

Parameters

<i>connection</i>	the serial connection to read a message from
-------------------	--

Returns

the readbuf if a message exists, else NULL

7.26.3.4 void serial_write (serial_t * connection, char * message)

Write a message to the serial communication link.

Parameters

<i>connection</i>	the serial communication link to write to
<i>message</i>	the message to send over to the other side

Note

be sure the message has a '
' character

7.27 SEVATBR-doc/robot/tbr/tbr.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <sys/types.h>
#include <dirent.h>
#include <termios.h>
#include <time.h>
#include "tbr.h"
```

Macros

- #define NUM_DEV 3
- #define DEV_BAUD B38400
- #define WHEEL_DEVID 1
- #define ARM_DEVID 2
- #define CLAW_DEVID 3
- #define SYNC_NSEC 500000000

Functions

- int tbr_connect (tbr_t *robot)
- void tbr_send (tbr_t *robot)
- void tbr_rcv (tbr_t *robot)
- void tbr_disconnect (tbr_t *robot)
- void tbr_reset (tbr_t *robot)

7.27.1 Macro Definition Documentation

7.27.1.1 #define ARM_DEVID 2

7.27.1.2 #define CLAW_DEVID 3

7.27.1.3 #define DEV_BAUD B38400

7.27.1.4 #define NUM_DEV 3

7.27.1.5 #define SYNC_NSEC 500000000

7.27.1.6 #define WHEEL_DEVID 1

7.27.2 Function Documentation

7.27.2.1 int tbr_connect (tbr_t * robot)

Initialize the communication layer

Parameters

<i>robot</i>	the robot information
--------------	-----------------------

Returns

number of devices initialized, -1 on error

7.27.2.2 void tbr_disconnect (tbr_t * robot)

Disconnect everything

Parameters

<i>robot</i>	the robot information
--------------	-----------------------

7.27.2.3 void tbr_recv (tbr_t * robot)

Receive input from the communication layer

Parameters

<i>robot</i>	the robot information
--------------	-----------------------

7.27.2.4 void tbr_reset (tbr_t * robot)

Reset the robot values

Parameters

<i>robot</i>	the robot information
--------------	-----------------------

7.27.2.5 void tbr_send (tbr_t * robot)

Send output to the communication layer

Parameters

<i>robot</i>	the robot information
--------------	-----------------------

7.28 SEVATBR-doc/robot/tbr/tbr.h File Reference

```
#include <stdint.h>
#include "serial.h"
```

Classes

- struct [tbr](#)

Typedefs

- typedef struct [tbr](#) [tbr_t](#)

Functions

- int [tbr_connect](#) ([tbr_t](#) *[robot](#))
- void [tbr_send](#) ([tbr_t](#) *[robot](#))
- void [tbr_recv](#) ([tbr_t](#) *[robot](#))
- void [tbr_disconnect](#) ([tbr_t](#) *[robot](#))
- void [tbr_reset](#) ([tbr_t](#) *[robot](#))

7.28.1 Typedef Documentation

7.28.1.1 typedef struct tbr tbr_t

7.28.2 Function Documentation

7.28.2.1 int tbr_connect (tbr_t * robot)

Initialize the communication layer

Parameters

<i>robot</i>	the robot information
--------------	-----------------------

Returns

number of devices initialized, -1 on error

7.28.2.2 void tbr_disconnect (tbr_t * robot)

Disconnect everything

Parameters

<i>robot</i>	the robot information
--------------	-----------------------

7.28.2.3 void tbr_recv (tbr_t * robot)

Receive input from the communication layer

Parameters

<i>robot</i>	the robot information
--------------	-----------------------

7.28.2.4 void tbr_reset (tbr_t * robot)

Reset the robot values

Parameters

<i>robot</i>	the robot information
--------------	-----------------------

7.28.2.5 void tbr_send (tbr_t * robot)

Send output to the communication layer

Parameters

<i>robot</i>	the robot information
--------------	-----------------------

7.29 SEVATBR-doc/slam/slam.c File Reference

```
#include <math.h>
#include "robot.h"
```

Functions

- void [set_noise](#) (slambot_t *[robot](#), double sensor_noise, double turn_noise, double motion_noise)
- void [move](#) (slambot_t *[robot](#), double theta, double radius)
- void [sense](#) (slambot_t *[robot](#))
- void [particle_update](#) (slambot_t *[robot](#), [pose3d_t](#) *trajectory)
- void [particle_resample](#) (slambot_t *[robot](#))
- double [gauss_error](#) (double mu, double sigma)

7.29.1 Function Documentation

7.29.1.1 double [gauss_error](#) (double *mu*, double *sigma*)

7.29.1.2 void [move](#) (slambot_t * *robot*, double *theta*, double *radius*)

7.29.1.3 void [particle_resample](#) (slambot_t * *robot*)

7.29.1.4 void [particle_update](#) (slambot_t * *robot*, [pose3d_t](#) * *trajectory*)

7.29.1.5 void [sense](#) (slambot_t * *robot*)

7.29.1.6 void [set_noise](#) (slambot_t * *robot*, double *sensor_noise*, double *turn_noise*, double *motion_noise*)

7.30 SEVATBR-doc/speech/speech-to-text/lse_stt.c File Reference

```
#include <stdio.h>
#include <stdint.h>
#include <stdlib.h>
#include <fcntl.h>
#include <string.h>
#include "lse_stt.h"
```

Classes

- union [note](#)

Macros

- #define [NTARGETS](#) 2

Functions

- static int16_t * [loadFile](#) (char *fname, long *s)
- static unsigned long long [square_error](#) (int16_t *haystack, int hlen, int16_t *needle, int nlen, int *start)
- int [stt_init](#) ([stt_t](#) *info)
- int [stt_decipher](#) ([stt_t](#) *info, char *fname, char **buf)
- void [stt_free](#) ([stt_t](#) *info)

Variables

- static char * [db](#) [NTARGETS]
- static long [length](#) [NTARGETS]
- static int16_t * [dbfiles](#) [NTARGETS]
- static char * [text](#) [NTARGETS]
- static char [goodtogo](#)

7.30.1 Macro Definition Documentation

7.30.1.1 #define NTARGETS 2

7.30.2 Function Documentation

7.30.2.1 int16_t* loadFile (char * *fname*, long * *s*) [static]

Load a file from a filename, and decode into 16000HZ, 16 bit signed little endian integers

Parameters

<i>fname</i>	name of the raw file
<i>s</i>	the changable size variable

Returns

the array (malloc'd) of notes, else NULL

7.30.2.2 unsigned long long square_error (int16_t * *haystack*, int *hlen*, int16_t * *needle*, int *nlen*, int * *start*) [static]

Get the least square error (lse) of the file to the dbfile comparison

Parameters

<i>haystack</i>	the dbfile
<i>hlen</i>	length of the haystack
<i>needle</i>	the file inputted for lse comparison
<i>nlen</i>	the length of the needle
<i>start</i>	the changable pointer to the beginning of the array

Returns

the lse on success, else max error (uint64_t)-1

7.30.2.3 int stt_decipher (stt_t * *info*, char * *fname*, char ** *buf*)

Decipher the current file for some hypothesis

Parameters

<i>info</i>	info for the lse_stt engine
<i>fname</i>	the name of the current file
<i>buf</i>	the buffer to store the deciphered hypothesis

Returns

length of the deciphered hypothesis on success, else -1

7.30.2.4 void stt_free (stt_t * *info*)

Free the current engine

Parameters

<i>info</i>	info for the lse_stt engine
-------------	-----------------------------

7.30.2.5 int stt_init (stt_t * info)

Initialize the speech engine for the lse matching.

Parameters

<i>info</i>	info for the lse_stt engine
-------------	-----------------------------

Returns

0 on success, -1 on error

7.30.3 Variable Documentation

7.30.3.1 char* db[NTARGETS] [static]

7.30.3.2 int16_t* dbfiles[NTARGETS] [static]

7.30.3.3 char goodtogo [static]

7.30.3.4 long length[NTARGETS] [static]

7.30.3.5 char* text[NTARGETS] [static]

7.31 SEVATBR-doc/speech/speech-to-text/lse_stt.h File Reference

Classes

- struct [stt](#)

Typedefs

- typedef struct [stt](#) [stt_t](#)

Functions

- int [stt_init](#) ([stt_t](#) *[info](#))
- int [stt_decipher](#) ([stt_t](#) *[info](#), char *[filename](#), char **[buf](#))
- void [stt_free](#) ([stt_t](#) *[info](#))

7.31.1 Typedef Documentation

7.31.1.1 typedef struct [stt](#) [stt_t](#)

7.31.2 Function Documentation

7.31.2.1 int stt_decipher (stt_t * info, char * filename, char ** buf)

Decipher the current file for some hypothesis

Parameters

<i>info</i>	info for the lse_stt engine
<i>fname</i>	the name of the current file
<i>buf</i>	the buffer to store the deciphered hypothesis

Returns

length of the deciphered hypothesis on success, else -1

Try to decipher a file

Parameters

<i>info</i>	the information struct for the engine
-------------	---------------------------------------

Returns

n characters deciphered on success, -1 otherwise

7.31.2.2 void stt_free (stt_t * info)

Free the current engine

Parameters

<i>info</i>	info for the lse_stt engine
-------------	-----------------------------

Remove the speech engine

Parameters

<i>info</i>	the information struct for the engine
-------------	---------------------------------------

7.31.2.3 int stt_init (stt_t * info)

Initialize the speech engine for the lse matching.

Parameters

<i>info</i>	info for the lse_stt engine
-------------	-----------------------------

Returns

0 on success, -1 on error

Initialize the speech engine

Parameters

<i>info</i>	the information struct for the engine
-------------	---------------------------------------

Returns

0 on success, -1 otherwise

7.32 SEVATBR-doc/speech/speech-to-text/rawrec.c File Reference

```
#include <gst/gst.h>
#include <stdlib.h>
#include <string.h>
#include "rawrec.h"
```

Functions

- int [start_recording](#) ([rawrec_t](#) *rr, char *fsinkloc)
- void [stop_recording](#) ([rawrec_t](#) *rr)

Variables

- static unsigned char [gst_initiallized](#)

7.32.1 Function Documentation**7.32.1.1 int start_recording (rawrec_t * rr, char * fsinkloc)**

Start the recording of the audio device.

Parameters

<i>rr</i>	the struct containing the necessary information for recording
<i>fsinkloc</i>	the filename of the destination

Returns

0 on success, -1 on error

7.32.1.2 void stop_recording (rawrec_t * rr)

Stop recording of the audio device.

Parameters

<i>rr</i>	the struct of the pipeline information
-----------	--

7.32.2 Variable Documentation**7.32.2.1 unsigned char gst_initiallized [static]****7.33 SEVATBR-doc/speech/speech-to-text/rawrec.h File Reference****Classes**

- struct [rawrec](#)

Typedefs

- typedef struct [rawrec](#) [rawrec_t](#)

Functions

- int [start_recording](#) ([rawrec_t](#) *rr, char *fsinkloc)
- void [stop_recording](#) ([rawrec_t](#) *rr)

7.33.1 Typedef Documentation

7.33.1.1 typedef struct rawrec rawrec_t

7.33.2 Function Documentation

7.33.2.1 int start_recording (rawrec_t * rr, char * fsinkloc)

Start the recording of the audio device.

Parameters

<i>rr</i>	the struct containing the necessary information for recording
<i>fsinkloc</i>	the filename of the destination

Returns

0 on success, -1 on error

7.33.2.2 void stop_recording (rawrec_t * rr)

Stop recording of the audio device.

Parameters

<i>rr</i>	the struct of the pipeline information
-----------	--

7.34 SEVATBR-doc/speech/speech-to-text/speech_signal.c File Reference

```
#include <pthread.h>
#include <unistd.h>
#include <sys/wait.h>
#include <sys/types.h>
#include <stdlib.h>
#include <signal.h>
#include <string.h>
#include <stdio.h>
#include <fcntl.h>
#include "rawrec.h"
#include "stt.h"
#include "speech_signal.h"
```

Macros

- #define `READPROG` "sample_raw.sh"

Functions

- static void * `_update_signals` (void *)
- int `start_speech_signals` (void)
- void `get_signal` (`speech_signal_t` *sigframe)
- void `stop_speech_signals` (void)

Variables

- static unsigned char `sslib_init`
- static unsigned char `sslib_exit`
- static `stt_t` `sre`
- static pthread_t `gst_manager`
- static pthread_mutex_t `gst_lock`
- static `speech_signal_t` `gst_sig`

7.34.1 Macro Definition Documentation

7.34.1.1 #define `READPROG` "sample_raw.sh"

7.34.2 Function Documentation

7.34.2.1 static void * `_update_signals` (void * *args*) [static]

Start two streams to record audio.

Parameters

<i>args</i>	Does nothing.
-------------	---------------

Returns

NULL

7.34.2.2 void `get_signal` (`speech_signal_t` * *sigframe*)

Gets the current state of signals for the sigframe

Parameters

<i>sigframe</i>	the sigframe to send over in order to copy the signals
-----------------	--

7.34.2.3 int `start_speech_signals` (void)

Start recording

7.34.2.4 void `stop_speech_signals` (void)

Stop recording

7.34.3 Variable Documentation

7.34.3.1 `pthread_mutex_t gst_lock` `[static]`

7.34.3.2 `pthread_t gst_manager` `[static]`

7.34.3.3 `speech_signal_t gst_sig` `[static]`

7.34.3.4 `stt_t sre` `[static]`

7.34.3.5 `unsigned char sslib_exit` `[static]`

7.34.3.6 `unsigned char sslib_init` `[static]`

7.35 SEVATBR-doc/speech/speech-to-text/speech_signal.h File Reference

Classes

- struct [speech_signal](#)

Typedefs

- typedef struct [speech_signal](#) [speech_signal_t](#)

Functions

- int [start_speech_signals](#) (void)
- void [get_signal](#) ([speech_signal_t](#) *sigframe)
- void [stop_speech_signals](#) (void)

7.35.1 Typedef Documentation

7.35.1.1 typedef struct [speech_signal](#) [speech_signal_t](#)

7.35.2 Function Documentation

7.35.2.1 void [get_signal](#) ([speech_signal_t](#) * *sigframe*)

Gets the current state of signals for the sigframe

Parameters

<i>sigframe</i>	the sigframe to send over in order to copy the signals
-----------------	--

7.35.2.2 int [start_speech_signals](#) (void)

Start recording

7.35.2.3 void [stop_speech_signals](#) (void)

Stop recording

7.36 SEVATBR-doc/speech/speech-to-text/stt.c File Reference

```
#include "stt.h"
#include <stdio.h>
#include <string.h>
#include <signal.h>
```

Functions

- int [stt_init](#) ([stt_t *info](#))
- int [stt_decipher](#) ([stt_t *info](#), char *filename, char **buf)
- void [stt_free](#) ([stt_t *info](#))

7.36.1 Function Documentation

7.36.1.1 int stt_decipher (stt_t * info, char * filename, char ** buf)

Try to decipher a file

Parameters

<i>info</i>	the information struct for the engine
-------------	---------------------------------------

Returns

n characters deciphered on success, -1 otherwise

7.36.1.2 void stt_free (stt_t * info)

Remove the speech engine

Parameters

<i>info</i>	the information struct for the engine
-------------	---------------------------------------

7.36.1.3 int stt_init (stt_t * info)

Initialize the speech engine

Parameters

<i>info</i>	the information struct for the engine
-------------	---------------------------------------

Returns

0 on success, -1 otherwise

7.37 SEVATBR-doc/speech/speech-to-text/stt.h File Reference

```
#include <pocketsphinx.h>
```


Classes

- struct [stt](#)

Typedefs

- typedef struct [stt stt_t](#)

Functions

- int [stt_init](#) ([stt_t *info](#))
- int [stt_decipher](#) ([stt_t *info](#), char *filename, char **buf)
- void [stt_free](#) ([stt_t *info](#))

7.37.1 Typedef Documentation

7.37.1.1 typedef struct stt stt_t

7.37.2 Function Documentation

7.37.2.1 int stt_decipher (stt_t * info, char * filename, char ** buf)

Decipher the current file for some hypothesis

Parameters

<i>info</i>	info for the lse_stt engine
<i>fname</i>	the name of the current file
<i>buf</i>	the buffer to store the deciphered hypothesis

Returns

length of the deciphered hypothesis on success, else -1

Try to decipher a file

Parameters

<i>info</i>	the information struct for the engine
-------------	---------------------------------------

Returns

n characters deciphered on success, -1 otherwise

7.37.2.2 void stt_free (stt_t * info)

Free the current engine

Parameters

<i>info</i>	info for the lse_stt engine
-------------	-----------------------------

Remove the speech engine

Parameters

<i>info</i>	the information struct for the engine
-------------	---------------------------------------

7.37.2.3 `int stt_init (stt_t * info)`

Initialize the speech engine for the lse matching.

Parameters

<i>info</i>	info for the lse_stt engine
-------------	-----------------------------

Returns

0 on success, -1 on error

Initialize the speech engine

Parameters

<i>info</i>	the information struct for the engine
-------------	---------------------------------------

Returns

0 on success, -1 otherwise

7.38 SEVATBR-doc/visual/basket.py File Reference

Namespaces

- [basket](#)

Functions

- `def basket._basket_image_hue_filter`
Internal wrapper image hue filter.
- `def basket._save_image`
Saves an image to the current directory.
- `def basket._init_particle_filter`
Internal wrapper to particle filter initializer.
- `def basket.is_basket_middle`
Single entry function returning True/False if basket is in the middle of the screen.
- `def basket.run_middle`
Runs continuously and prints if the best detected blob is in the middle.
- `def basket.run`
Runs continuously outlines best matched blob if it is in the middle.

Variables

- `basket.particle_filter = None`
- `int basket.image_half_size = -1`
- `int basket.save_count = 1`
- `tuple basket.base_filename = datetime.now()`

7.39 SEVATBR-doc/visual/basket_runner.py File Reference

Namespaces

- [basket_runner](#)

7.40 SEVATBR-doc/visual/basket_test.py File Reference

Namespaces

- [basket_test](#)

Functions

- def [basket_test.unitTest](#)
- def [basket_test.basketPresent](#)
- def [basket_test.basketMissing](#)

7.41 SEVATBR-doc/visual/experiment.py File Reference

Namespaces

- [experiment](#)

Functions

- def [experiment.experiment](#)
- def [experiment.hard_threshold](#)
- def [experiment.binary_mask](#)
- def [experiment.dilation_and_blur](#)
- def [experiment.blobs_by_mask](#)

7.42 SEVATBR-doc/visual/image_support.py File Reference

Namespaces

- [image_support](#)

Functions

- def [image_support.external_init_particle_filter](#)
Initializes particle filter.
- def [image_support.image_hue_filter](#)
Converts given image to HSV based on the given color.
- def [image_support.get_hue_blobs](#)
Gets basket blobs after hue distance filtering.
- def [image_support.get_best_blob](#)
Returns the best blob out of the provided set and particle filter.
- def [image_support.is_blob_in_middle_helper](#)
Determines whether the given blob is in ceter of image.

7.43 SEVATBR-doc/visual/particlefilter.py File Reference

Classes

- class [particlefilter.ParticleFilter](#)

Namespaces

- [particlefilter](#)

7.44 SEVATBR-doc/visual/prquadtree.py File Reference

Classes

- class [prquadtree.Point](#)
Represents an (x,y) coordinate point on a grid.
- class [prquadtree.Particle](#)
Represents particle point.
- class [prquadtree.Box](#)
Class defining a square on the coordinate system via a center point and half of square width.
- class [prquadtree.PRQuadTree](#)
Class representing a [Point](#) Range Quadtree.

Namespaces

- [prquadtree](#)

7.45 SEVATBR-doc/visual/prquadtree_test.py File Reference

Classes

- class [prquadtree_test.TestPoint](#)
- class [prquadtree_test.TestParticle](#)
- class [prquadtree_test.TestBox](#)
- class [prquadtree_test.TestPrQuadTree](#)

Namespaces

- [prquadtree_test](#)

7.46 SEVATBR-doc/visual/prquadtree_test_example.py File Reference

Namespaces

- [prquadtree_test_example](#)

Variables

- tuple `prquadtree_test_example.b` = `Box(Point(5,5), 50)`
- tuple `prquadtree_test_example.b2` = `Box(Point(50,50), 50)`
- tuple `prquadtree_test_example.qt` = `PRQuadTree(b2)`
- tuple `prquadtree_test_example.pt` = `Point(2,2)`
- tuple `prquadtree_test_example.nearby` = `qt.query_k_nearest(pt, 20)`
- int `prquadtree_test_example.c` = 1

7.47 SEVATBR-doc/visual/tennis_ball.py File Reference

Namespaces

- `tennis_ball`

Functions

- def `tennis_ball._init_particle_filter`
Internal wrapper to particle filter initializer.
- def `tennis_ball._ball_image_hue_filter`
Internal wrapper image hue filter.
- def `tennis_ball.is_ball_middle`
Entry point for module which determines whether tennis ball is in the middle of the image.
- def `tennis_ball.run`
Continuously captures image from computer camera and feeds it to the `is_ball_middle` method to detect whether tennis ball is in the middle of the screen.

Variables

- `tennis_ball.particle_filter` = None

7.48 SEVATBR-doc/visual/tennis_ball_runner.py File Reference

Namespaces

- `tennis_ball_runner`

7.49 SEVATBR-doc/visual/tennis_ball_test.py File Reference

Namespaces

- `tennis_ball_test`

Functions

- def `tennis_ball_test.unitTest`
- def `tennis_ball_test.ballPresent`
- def `tennis_ball_test.ballMissing`

7.50 SEVATBR-doc/visual/visual.h File Reference

Functions

- int [start_visual](#) (void)
- void [set_objects](#) (object_t *objs)
- void [get_objects](#) (object_t *objs, [point_t](#) *locations)
- void [stop_visual](#) (void)

7.50.1 Function Documentation

7.50.1.1 void [get_objects](#) (object_t * *objs*, point_t * *locations*)

7.50.1.2 void [set_objects](#) (object_t * *objs*)

7.50.1.3 int [start_visual](#) (void)

7.50.1.4 void [stop_visual](#) (void)

Index

- `__init__`
 - `particlefilter::ParticleFilter`, 26
 - `prquadtree::Box`, 19
 - `prquadtree::Particle`, 25
 - `prquadtree::Point`, 27
 - `prquadtree::PRQuadTree`, 30
- `__repr__`
 - `prquadtree::Point`, 27
- `__str__`
 - `prquadtree::Point`, 27
 - `prquadtree::PRQuadTree`, 31
- `_ball_image_hue_filter`
 - `tennis_ball`, 15
- `_basket_image_hue_filter`
 - `basket`, 9
- `_ctrl`, 19
- `_init_particle_filter`
 - `basket`, 9
 - `tennis_ball`, 16
- `_save_image`
 - `basket`, 10
- `_serial_setattr`
 - `serial.c`, 67
- `_serial_update`
 - `serial.c`, 67
- `_subdivide`
 - `prquadtree::PRQuadTree`, 31
- `_update_signals`
 - `speech_signal.c`, 82

A

- `controller`, 21
- `AGENT_SIMPLE`
 - `agent.h`, 41
- `ARM_DEVID`
 - `tbr.c`, 72
- `ARM_L1`
 - `arm.cpp`, 60
- `ARM_L2`
 - `arm.cpp`, 60
- `ARM_L3`
 - `arm.cpp`, 60
- `ARM_R1`
 - `arm.cpp`, 60
- `ARM_R2`
 - `arm.cpp`, 60
- `ARM_R3`
 - `arm.cpp`, 60
- `agent.c`
 - `conscious_thought`, 39

- `consciousness`, 41
- `ctrl`, 41
- `end_consciousness`, 41
- `gotosleep`, 41
- `info_t`, 39
- `wakeup`, 41
- `agent.h`
 - `AGENT_SIMPLE`, 41
 - `agent_create`, 41
 - `agent_destroy`, 42
 - `agent_disable`, 42
 - `agent_enable`, 42
 - `agent_get_poses`, 42
- `agent/test.c`
 - `exit_signal`, 45
 - `main`, 45
 - `stopprog`, 45
- `agent_create`
 - `agent.h`, 41
 - `simple_agent.cpp`, 43
- `agent_destroy`
 - `agent.h`, 42
 - `simple_agent.cpp`, 44
- `agent_disable`
 - `agent.h`, 42
 - `simple_agent.cpp`, 44
- `agent_enable`
 - `agent.h`, 42
 - `simple_agent.cpp`, 44
- `agent_enabled`
 - `simple_agent.cpp`, 44
- `agent_get_poses`
 - `agent.h`, 42
 - `simple_agent.cpp`, 44
- `arm`
 - `manual.c`, 58
 - `tbr`, 35
- `arm.cpp`
 - `ARM_L1`, 60
 - `ARM_L2`, 60
 - `ARM_L3`, 60
 - `ARM_R1`, 60
 - `ARM_R2`, 60
 - `ARM_R3`, 60
 - `arm_l`, 61
 - `arm_r`, 61
 - `DEV_ID`, 60
 - `limit`, 60
 - `loop`, 60

- lowerarm, 61
- msg, 61
- raisearm, 61
- setarm, 61
- setup, 61
- stoparm, 61
- time, 61
- arm_l
 - arm.cpp, 61
- arm_r
 - arm.cpp, 61
- attach
 - HBridgeMotor, 22
- axes
 - controller, 21
- B
 - controller, 21
- b
 - prquadtree_test_example, 15
- b2
 - prquadtree_test_example, 15
- BOT_L
 - claw.cpp, 62
- BOT_R
 - claw.cpp, 62
- backward
 - wheels.cpp, 64
- ballMissing
 - tennis_ball_test, 16
- ballPresent
 - tennis_ball_test, 16
- base
 - manual.c, 58
- base_filename
 - basket, 10
- basket, 9
 - _basket_image_hue_filter, 9
 - _init_particle_filter, 9
 - _save_image, 10
 - base_filename, 10
 - image_half_size, 10
 - is_basket_middle, 10
 - particle_filter, 10
 - run, 10
 - run_middle, 10
 - save_count, 10
- basket_runner, 10
- basket_test, 10
 - basketMissing, 11
 - basketPresent, 11
 - unitTest, 11
- basketMissing
 - basket_test, 11
- basketPresent
 - basket_test, 11
- baudrate
 - serial, 34
- binary_mask
 - experiment, 11
- blobs_by_mask
 - experiment, 11
- box
 - prquadtree::PRQuadTree, 32
- buffer
 - rawrec, 33
 - serial, 34
- buttons
 - controller, 21
- c
 - prquadtree_test_example, 15
- CLAW_DEVID
 - tbr.c, 72
- CLAW_L
 - claw.cpp, 62
- CLAW_R
 - claw.cpp, 62
- caps
 - rawrec, 33
- center
 - prquadtree::Box, 20
- claw
 - tbr, 35
- claw.cpp
 - BOT_L, 62
 - BOT_R, 62
 - CLAW_L, 62
 - CLAW_R, 62
 - claw_l, 62
 - claw_r, 62
 - closeclaw, 62
 - DEV_ID, 62
 - limit, 62
 - loop, 62
 - msg, 62
 - openclaw, 62
 - setclaw, 62
 - setup, 62
 - stopclaw, 62
 - TOP_L, 62
 - TOP_R, 62
 - time, 62
- claw_l
 - claw.cpp, 62
- claw_r
 - claw.cpp, 62
- clear_scores
 - particlefilter::ParticleFilter, 26
- closeclaw
 - claw.cpp, 62
- conscious_thought
 - agent.c, 39
- consciousness
 - agent.c, 41
- config
 - stt, 35
- connected

- controller, 21
 - httplink, 23
 - serial, 34
 - tbr, 35
- connections
 - tbr, 35
- contains_point
 - prquadtree::Box, 20
- controller, 20
 - A, 21
 - axes, 21
 - B, 21
 - buttons, 21
 - connected, 21
 - DOWN, 21
 - fd, 21
 - HOME, 21
 - LB, 21
 - LB2, 21
 - LEFT, 21
 - LJOY, 21
 - LT, 21
 - name, 21
 - pressed, 22
 - RB, 22
 - RB2, 22
 - RIGHT, 22
 - RJOY, 22
 - RT, 22
 - SELECT, 22
 - START, 22
 - UP, 22
 - X, 22
 - x, 22
 - Y, 22
 - y, 22
- controller.c
 - controller_connect, 50
 - controller_disconnect, 50
 - controller_update, 52
 - INPUT_DIR, 50
 - JS_PREFIX, 50
 - MAX_16BIT, 50
- controller.h
 - controller_connect, 52
 - controller_disconnect, 52
 - controller_t, 52
- controller_connect
 - controller.c, 50
 - controller.h, 52
- controller_disconnect
 - controller.c, 50
 - controller.h, 52
- controller_t
 - controller.h, 52
- controller_update
 - controller.c, 52
 - manual.c, 56
- coord.h
 - point2d_t, 46
 - point3d_t, 46
 - point_t, 46
 - pose2d_t, 46
 - pose3d_t, 47
 - pose_t, 47
- core.c
 - main, 47
 - stop_program, 47
 - stop_signal, 47
- core.cpp
 - main, 48
 - stop_program, 48
 - stop_signal, 48
- ctrl
 - agent.c, 41
 - manual.c, 58
- currid
 - robot.c, 65
- DEV_BAUD
 - tbr.c, 72
- DEV_ID
 - arm.cpp, 60
 - claw.cpp, 62
 - wheels.cpp, 63
- DOWN
 - controller, 21
- db
 - lse_stt.c, 78
- dbfiles
 - lse_stt.c, 78
- dilation_and_blur
 - experiment, 11
- end_consciousness
 - agent.c, 41
- exit_signal
 - agent/test.c, 45
- experiment, 11
 - binary_mask, 11
 - blobs_by_mask, 11
 - dilation_and_blur, 11
 - experiment, 11
 - hard_threshold, 11
- external_init_particle_filter
 - image_support, 12
- fd
 - controller, 21
 - serial, 34
- fetch
 - speech_signal, 34
- filesink_loc
 - rawrec, 33
- forward
 - wheels.cpp, 64

- gauss_error
 - slam.c, 75
- get_best_blob
 - image_support, 13
- get_hue_blobs
 - image_support, 13
- get_objects
 - visual.h, 90
- get_signal
 - speech_signal.c, 82
 - speech_signal.h, 83
- go
 - speech_signal, 34
- goodtogo
 - lse_stt.c, 78
- gotosleep
 - agent.c, 41
- gst_initialized
 - rawrec.c, 80
- gst_lock
 - speech_signal.c, 83
- gst_manager
 - speech_signal.c, 83
- gst_sig
 - speech_signal.c, 83
- HBridgeMotor, 22
 - attach, 22
 - HBridgeMotor, 22
 - HBridgeMotor, 22
 - isdigital, 23
 - pin, 23
 - reset, 23
 - setdigital, 23
 - velocity, 23
 - write, 23
- HOME
 - controller, 21
- HZ
 - manual.c, 56
- half_size
 - prquadtree::Box, 20
- hard_threshold
 - experiment, 11
- high
 - note, 24
- hostname
 - httplink, 23
- httplink, 23
 - connected, 23
 - hostname, 23
 - ipaddr, 23
 - socket_fd, 23
- httplink.c
 - httplink_connect, 53
 - httplink_disconnect, 53
 - httplink_recv, 53
 - httplink_send, 54
 - msgbuf, 54
 - response, 54
- httplink.h
 - httplink_connect, 54
 - httplink_disconnect, 55
 - httplink_recv, 55
 - httplink_send, 55
 - httplink_t, 54
- httplink_connect
 - httplink.c, 53
 - httplink.h, 54
- httplink_disconnect
 - httplink.c, 53
 - httplink.h, 55
- httplink_recv
 - httplink.c, 53
 - httplink.h, 55
- httplink_send
 - httplink.c, 54
 - httplink.h, 55
- httplink_t
 - httplink.h, 54
- INPUT_DIR
 - controller.c, 50
 - serial.c, 67
- ids
 - tbr, 35
- image_box
 - particlefilter::ParticleFilter, 26
- image_half_size
 - basket, 10
- image_hue_filter
 - image_support, 13
- image_support, 11
 - external_init_particle_filter, 12
 - get_best_blob, 13
 - get_hue_blobs, 13
 - image_hue_filter, 13
 - is_blob_in_middle_helper, 13
- info, 23
 - ss, 24
- info_t
 - agent.c, 39
- input_id
 - manual.c, 58
- insert
 - prquadtree::PRQuadTree, 31
- intersect
 - prquadtree::Box, 20
- ipaddr
 - httplink, 23
- is_ball_middle
 - tennis_ball, 16
- is_basket_middle
 - basket, 10
- is_blob_in_middle_helper
 - image_support, 13
- isOverriden
 - manual.c, 56

- manual.h, 58
- isdigital
 - HBridgeMotor, 23
- iterate
 - particlefilter::ParticleFilter, 26
- iterations
 - particlefilter::ParticleFilter, 26
- iterations_before_clearing
 - particlefilter::ParticleFilter, 26
- JS_PREFIX
 - controller.c, 50
- LB
 - controller, 21
- LB2
 - controller, 21
- LEFT
 - controller, 21
- LJOY
 - controller, 21
- LT
 - controller, 21
- last_signal
 - manual.c, 58
 - simple_agent.cpp, 44
- left
 - tbr, 35
- length
 - lse_stt.c, 78
- limit
 - arm.cpp, 60
 - claw.cpp, 62
 - wheels.cpp, 64
- loadFile
 - lse_stt.c, 76
- log_destroy
 - logger.h, 50
- log_init
 - logger.h, 50
- log_print
 - logger.h, 50
- log_scan
 - logger.h, 50
- logfile
 - logger.c, 49
- logger.c
 - logfile, 49
 - logger_destroy, 49
 - logger_init, 49
 - logger_print, 49
 - logger_scan, 49
- logger.h
 - log_destroy, 50
 - log_init, 50
 - log_print, 50
 - log_scan, 50
- logger_destroy
 - logger.c, 49
- logger_init
 - logger.c, 49
- logger_print
 - logger.c, 49
- logger_scan
 - logger.c, 49
- loop
 - arm.cpp, 60
 - claw.cpp, 62
 - wheels.cpp, 64
- low
 - note, 24
- lowerarm
 - arm.cpp, 61
- lse_stt.c
 - db, 78
 - dbfiles, 78
 - goodtogo, 78
 - length, 78
 - loadFile, 76
 - NTARGETS, 76
 - square_error, 76
 - stt_decipher, 76
 - stt_free, 76
 - stt_init, 78
 - text, 78
- lse_stt.h
 - stt_decipher, 78
 - stt_free, 79
 - stt_init, 79
 - stt_t, 78
- MAX_16BIT
 - controller.c, 50
- MNL_CTRL
 - manual.h, 58
- MNL_SRVR
 - manual.h, 58
- main
 - agent/test.c, 45
 - core.c, 47
 - core.cpp, 48
 - manual/test.c, 45
 - robot/tbr/test.c, 45
 - speech/speech-to-text/test.c, 46
- manual.c
 - arm, 58
 - base, 58
 - controller_update, 56
 - ctrl, 58
 - HZ, 56
 - input_id, 58
 - isOverriden, 56
 - last_signal, 58
 - manual_connect, 56
 - manual_disable, 57
 - manual_disconnect, 57
 - manual_en, 58
 - manual_enable, 57

- manual_get_poses, [57](#)
 - manual_new_data, [57](#)
 - mnl_override, [58](#)
 - new_join, [58](#)
 - raise_server_request, [57](#)
 - server, [58](#)
 - server_update, [57](#)
- manual.h
 - isOverriden, [58](#)
 - MNL_CTRL, [58](#)
 - MNL_SRVR, [58](#)
 - manual_connect, [59](#)
 - manual_disable, [59](#)
 - manual_disconnect, [59](#)
 - manual_enable, [59](#)
 - manual_get_poses, [59](#)
 - manual_new_data, [59](#)
- manual/test.c
 - main, [45](#)
 - stop, [45](#)
 - stopsig, [45](#)
- manual_connect
 - manual.c, [56](#)
 - manual.h, [59](#)
- manual_disable
 - manual.c, [57](#)
 - manual.h, [59](#)
- manual_disconnect
 - manual.c, [57](#)
 - manual.h, [59](#)
- manual_en
 - manual.c, [58](#)
- manual_enable
 - manual.c, [57](#)
 - manual.h, [59](#)
- manual_get_poses
 - manual.c, [57](#)
 - manual.h, [59](#)
- manual_new_data
 - manual.c, [57](#)
 - manual.h, [59](#)
- mnl_override
 - manual.c, [58](#)
- move
 - slam.c, [75](#)
- msg
 - arm.cpp, [61](#)
 - claw.cpp, [62](#)
 - wheels.cpp, [64](#)
- msgbuf
 - httplink.c, [54](#)
- NO_ROBOT
 - robot.h, [66](#)
- NTARGETS
 - lse_stt.c, [76](#)
- NUM_DEV
 - tbr.c, [72](#)
- name
 - controller, [21](#)
- ne
 - prquadtree::PRQuadTree, [32](#)
- nearby
 - prquadtree_test_example, [15](#)
- new_join
 - manual.c, [58](#)
- none
 - speech_signal, [34](#)
- note, [24](#)
 - high, [24](#)
 - low, [24](#)
 - val, [24](#)
- nothing
 - stt, [35](#)
- num_possible
 - tbr, [35](#)
- nw
 - prquadtree::PRQuadTree, [32](#)
- openclaw
 - claw.cpp, [62](#)
- PREFIXES
 - serial.c, [69](#)
- parity
 - serial, [34](#)
- particle_filter
 - basket, [10](#)
 - tennis_ball, [16](#)
- particle_resample
 - slam.c, [75](#)
- particle_update
 - slam.c, [75](#)
- particlefilter, [14](#)
- particlefilter.ParticleFilter, [25](#)
- particlefilter::ParticleFilter
 - __init__, [26](#)
 - clear_scores, [26](#)
 - image_box, [26](#)
 - iterate, [26](#)
 - iterations, [26](#)
 - iterations_before_clearing, [26](#)
 - pr_tree, [26](#)
 - score, [26](#)
- pin
 - HBridgeMotor, [23](#)
- pipeline
 - rawrec, [33](#)
- pitch
 - pose3d, [29](#)
- point2d, [28](#)
 - x, [28](#)
 - y, [28](#)
- point2d_t
 - coord.h, [46](#)
- point3d, [28](#)
 - x, [28](#)
 - y, [28](#)

- z, [28](#)
- point3d_t
 - coord.h, [46](#)
- point_t
 - coord.h, [46](#)
- points
 - prquadtree::PRQuadTree, [32](#)
- port
 - serial, [34](#)
- pose2d, [29](#)
 - theta, [29](#)
 - x, [29](#)
 - y, [29](#)
- pose2d_t
 - coord.h, [46](#)
- pose3d, [29](#)
 - pitch, [29](#)
 - roll, [29](#)
 - x, [29](#)
 - y, [29](#)
 - yaw, [29](#)
 - z, [29](#)
- pose3d_t
 - coord.h, [47](#)
- pose_t
 - coord.h, [47](#)
- possible_ports
 - tbr, [36](#)
- pr_tree
 - particlefilter::ParticleFilter, [26](#)
- pressed
 - controller, [22](#)
- prev_arm
 - tbr, [36](#)
- prev_claw
 - tbr, [36](#)
- prev_left
 - tbr, [36](#)
- prev_right
 - tbr, [36](#)
- print_all_points
 - prquadtree::PRQuadTree, [31](#)
- prquadtree, [14](#)
 - prquadtree.Box, [19](#)
 - prquadtree.PRQuadTree, [30](#)
 - prquadtree.Particle, [24](#)
 - prquadtree.Point, [26](#)
 - prquadtree::Box
 - __init__, [19](#)
 - center, [20](#)
 - contains_point, [20](#)
 - half_size, [20](#)
 - intersect, [20](#)
 - prquadtree::PRQuadTree
 - __init__, [30](#)
 - __str__, [31](#)
 - _subdivide, [31](#)
 - box, [32](#)
 - insert, [31](#)
 - ne, [32](#)
 - nw, [32](#)
 - points, [32](#)
 - print_all_points, [31](#)
 - query_k_nearest, [31](#)
 - query_range, [32](#)
 - se, [32](#)
 - size, [32](#)
 - sw, [32](#)
 - prquadtree::Particle
 - __init__, [25](#)
 - score, [25](#)
 - x, [25](#)
 - y, [25](#)
 - prquadtree::Point
 - __init__, [27](#)
 - __repr__, [27](#)
 - __str__, [27](#)
 - x, [28](#)
 - y, [28](#)
 - prquadtree_test, [14](#)
 - prquadtree_test.TestBox, [36](#)
 - prquadtree_test.TestParticle, [36](#)
 - prquadtree_test.TestPoint, [37](#)
 - prquadtree_test.TestPrQuadTree, [37](#)
 - prquadtree_test::TestBox
 - test_box_contains, [36](#)
 - test_box_insert, [36](#)
 - prquadtree_test::TestParticle
 - test_particle_insert, [37](#)
 - prquadtree_test::TestPoint
 - test_point_insert, [37](#)
 - prquadtree_test::TestPrQuadTree
 - test_insert, [38](#)
 - test_nearby, [38](#)
 - prquadtree_test_example, [14](#)
 - b, [15](#)
 - b2, [15](#)
 - c, [15](#)
 - nearby, [15](#)
 - pt, [15](#)
 - qt, [15](#)
 - ps
 - stt, [35](#)
 - pt
 - prquadtree_test_example, [15](#)
- QT_NODE_CAPACITY
 - prquadtree::PRQuadTree, [32](#)
- qt
 - prquadtree_test_example, [15](#)
- query_k_nearest
 - prquadtree::PRQuadTree, [31](#)
- query_range
 - prquadtree::PRQuadTree, [32](#)
- RB
 - controller, [22](#)

- RB2
 - controller, 22
- READPROG
 - speech_signal.c, 82
- RIGHT
 - controller, 22
- RJOY
 - controller, 22
- RT
 - controller, 22
- raise_server_request
 - manual.c, 57
- raisearm
 - arm.cpp, 61
- rawrec, 33
 - buffer, 33
 - caps, 33
 - filesink_loc, 33
 - pipeline, 33
 - sink, 33
 - source, 33
- rawrec.c
 - gst_initialized, 80
 - start_recording, 80
 - stop_recording, 80
- rawrec.h
 - rawrec_t, 81
 - start_recording, 81
 - stop_recording, 81
- rawrec_t
 - rawrec.h, 81
- readAvailable
 - serial, 34
- readbuf
 - serial, 34
- reset
 - HBridgeMotor, 23
- response
 - httplink.c, 54
- ret
 - speech_signal, 34
- right
 - tbr, 36
- robot
 - robot.c, 65
- robot.c
 - currid, 65
 - robot, 65
 - robot_move, 65
 - robot_set, 65
 - robot_unset, 65
- robot.h
 - NO_ROBOT, 66
 - robot_move, 66
 - robot_set, 66
 - robot_unset, 66
 - STANDARD_OUT, 66
 - TACHIKOMA, 66
 - TENNIS_BALL_ROBOT, 66
- robot/tbr/test.c
 - main, 45
- robot_move
 - robot.c, 65
 - robot.h, 66
- robot_set
 - robot.c, 65
 - robot.h, 66
- robot_unset
 - robot.c, 65
 - robot.h, 66
- roll
 - pose3d, 29
- run
 - basket, 10
 - tennis_ball, 16
- run_middle
 - basket, 10
- SELECT
 - controller, 22
- SEVATBR-doc/agent/agent.c, 39
- SEVATBR-doc/agent/agent.h, 41
- SEVATBR-doc/agent/simple_agent.cpp, 42
- SEVATBR-doc/agent/test.c, 44
- SEVATBR-doc/core/coord.h, 46
- SEVATBR-doc/core/core.c, 47
- SEVATBR-doc/core/core.cpp, 48
- SEVATBR-doc/logger/logger.c, 48
- SEVATBR-doc/logger/logger.h, 49
- SEVATBR-doc/manual/controller.c, 50
- SEVATBR-doc/manual/controller.h, 52
- SEVATBR-doc/manual/httplink.c, 53
- SEVATBR-doc/manual/httplink.h, 54
- SEVATBR-doc/manual/manual.c, 55
- SEVATBR-doc/manual/manual.h, 58
- SEVATBR-doc/manual/rtsp/rtsplink.c, 59
- SEVATBR-doc/manual/test.c, 45
- SEVATBR-doc/robot/arduino/arm/arm.cpp, 60
- SEVATBR-doc/robot/arduino/claw/claw.cpp, 61
- SEVATBR-doc/robot/arduino/wheels/wheels.cpp, 62
- SEVATBR-doc/robot/robot.c, 64
- SEVATBR-doc/robot/robot.h, 65
- SEVATBR-doc/robot/serial/serial.c, 67
- SEVATBR-doc/robot/serial/serial.h, 69
- SEVATBR-doc/robot/tbr/tbr.c, 71
- SEVATBR-doc/robot/tbr/tbr.h, 73
- SEVATBR-doc/robot/tbr/test.c, 45
- SEVATBR-doc/slam/slam.c, 74
- SEVATBR-doc/speech/speech-to-text/lse_stt.c, 75
- SEVATBR-doc/speech/speech-to-text/lse_stt.h, 78
- SEVATBR-doc/speech/speech-to-text/rawrec.c, 80
- SEVATBR-doc/speech/speech-to-text/rawrec.h, 80
- SEVATBR-doc/speech/speech-to-text/speech_signal.c, 81
- SEVATBR-doc/speech/speech-to-text/speech_signal.h, 83
- SEVATBR-doc/speech/speech-to-text/stt.c, 84

- SEVATBR-doc/speech/speech-to-text/stt.h, 84
- SEVATBR-doc/speech/speech-to-text/test.c, 46
- SEVATBR-doc/visual/basket.py, 86
- SEVATBR-doc/visual/basket_runner.py, 87
- SEVATBR-doc/visual/basket_test.py, 87
- SEVATBR-doc/visual/experiment.py, 87
- SEVATBR-doc/visual/image_support.py, 87
- SEVATBR-doc/visual/particlefilter.py, 88
- SEVATBR-doc/visual/prquadtree.py, 88
- SEVATBR-doc/visual/prquadtree_test.py, 88
- SEVATBR-doc/visual/prquadtree_test_example.py, 88
- SEVATBR-doc/visual/tennis_ball.py, 89
- SEVATBR-doc/visual/tennis_ball_runner.py, 89
- SEVATBR-doc/visual/tennis_ball_test.py, 89
- SEVATBR-doc/visual/visual.h, 90
- STANDARD_OUT
 - robot.h, 66
- START
 - controller, 22
- SWBUFMAX
 - serial.h, 69
- SWREADMAX
 - serial.h, 69
- SWWRITEMAX
 - serial.h, 69
- SYNC_NSEC
 - tbr.c, 72
- save_count
 - basket, 10
- score
 - particlefilter::ParticleFilter, 26
 - prquadtree::Particle, 25
- se
 - prquadtree::PRQuadTree, 32
- sense
 - slam.c, 75
- serial, 33
 - baudrate, 34
 - buffer, 34
 - connected, 34
 - fd, 34
 - parity, 34
 - port, 34
 - readAvailable, 34
 - readbuf, 34
- serial.c
 - _serial_setattr, 67
 - _serial_update, 67
 - INPUT_DIR, 67
 - PREFIXES, 69
 - serial_connect, 68
 - serial_disconnect, 68
 - serial_read, 68
 - serial_write, 68
 - tempbuf, 69
- serial.h
 - SWBUFMAX, 69
 - SWREADMAX, 69
 - SWWRITEMAX, 69
 - serial_connect, 70
 - serial_disconnect, 71
 - serial_read, 71
 - serial_t, 69
 - serial_write, 71
- serial_connect
 - serial.c, 68
 - serial.h, 70
- serial_disconnect
 - serial.c, 68
 - serial.h, 71
- serial_read
 - serial.c, 68
 - serial.h, 71
- serial_t
 - serial.h, 69
- serial_write
 - serial.c, 68
 - serial.h, 71
- server
 - manual.c, 58
- server_update
 - manual.c, 57
- set_noise
 - slam.c, 75
- set_objects
 - visual.h, 90
- setarm
 - arm.cpp, 61
- setclaw
 - claw.cpp, 62
- setdigital
 - HBridgeMotor, 23
- setup
 - arm.cpp, 61
 - claw.cpp, 62
 - wheels.cpp, 64
- setwheels
 - wheels.cpp, 64
- simple_agent.cpp
 - agent_create, 43
 - agent_destroy, 44
 - agent_disable, 44
 - agent_enable, 44
 - agent_enabled, 44
 - agent_get_poses, 44
 - last_signal, 44
 - view, 44
- sink
 - rawrec, 33
- size
 - prquadtree::PRQuadTree, 32
- slam.c
 - gauss_error, 75
 - move, 75
 - particle_resample, 75
 - particle_update, 75

- sense, 75
- set_noise, 75
- socket_fd
 - httplink, 23
- source
 - rawrec, 33
- speech/speech-to-text/test.c
 - main, 46
 - stop, 46
 - stopsig, 46
- speech_signal, 34
 - fetch, 34
 - go, 34
 - none, 34
 - ret, 34
 - stop, 34
- speech_signal.c
 - _update_signals, 82
 - get_signal, 82
 - gst_lock, 83
 - gst_manager, 83
 - gst_sig, 83
 - READPROG, 82
 - sre, 83
 - sslib_exit, 83
 - sslib_init, 83
 - start_speech_signals, 82
 - stop_speech_signals, 82
- speech_signal.h
 - get_signal, 83
 - speech_signal_t, 83
 - start_speech_signals, 83
 - stop_speech_signals, 83
- speech_signal_t
 - speech_signal.h, 83
- square_error
 - lse_stt.c, 76
- sre
 - speech_signal.c, 83
- ss
 - info, 24
- sslib_exit
 - speech_signal.c, 83
- sslib_init
 - speech_signal.c, 83
- start_recording
 - rawrec.c, 80
 - rawrec.h, 81
- start_speech_signals
 - speech_signal.c, 82
 - speech_signal.h, 83
- start_visual
 - visual.h, 90
- stop
 - manual/test.c, 45
 - speech/speech-to-text/test.c, 46
 - speech_signal, 34
- stop_program
 - core.c, 47
 - core.cpp, 48
- stop_recording
 - rawrec.c, 80
 - rawrec.h, 81
- stop_signal
 - core.c, 47
 - core.cpp, 48
- stop_speech_signals
 - speech_signal.c, 82
 - speech_signal.h, 83
- stop_visual
 - visual.h, 90
- stoparm
 - arm.cpp, 61
- stopclaw
 - claw.cpp, 62
- stopprog
 - agent/test.c, 45
- stopsig
 - manual/test.c, 45
 - speech/speech-to-text/test.c, 46
- stopwheels
 - wheels.cpp, 64
- stt, 34
 - config, 35
 - nothing, 35
 - ps, 35
- stt.c
 - stt_decipher, 84
 - stt_free, 84
 - stt_init, 84
- stt.h
 - stt_decipher, 85
 - stt_free, 85
 - stt_init, 86
 - stt_t, 85
- stt_decipher
 - lse_stt.c, 76
 - lse_stt.h, 78
 - stt.c, 84
 - stt.h, 85
- stt_free
 - lse_stt.c, 76
 - lse_stt.h, 79
 - stt.c, 84
 - stt.h, 85
- stt_init
 - lse_stt.c, 78
 - lse_stt.h, 79
 - stt.c, 84
 - stt.h, 86
- stt_t
 - lse_stt.h, 78
 - stt.h, 85
- sw
 - prquadtree::PRQuadTree, 32
- TACHIKOMA

- robot.h, 66
- TENNIS_BALL_ROBOT
 - robot.h, 66
- TOP_L
 - claw.cpp, 62
- TOP_R
 - claw.cpp, 62
- tbr, 35
 - arm, 35
 - claw, 35
 - connected, 35
 - connections, 35
 - ids, 35
 - left, 35
 - num_possible, 35
 - possible_ports, 36
 - prev_arm, 36
 - prev_claw, 36
 - prev_left, 36
 - prev_right, 36
 - right, 36
- tbr.c
 - ARM_DEVID, 72
 - CLAW_DEVID, 72
 - DEV_BAUD, 72
 - NUM_DEV, 72
 - SYNC_NSEC, 72
 - tbr_connect, 72
 - tbr_disconnect, 72
 - tbr_recv, 73
 - tbr_reset, 73
 - tbr_send, 73
 - WHEEL_DEVID, 72
- tbr.h
 - tbr_connect, 74
 - tbr_disconnect, 74
 - tbr_recv, 74
 - tbr_reset, 74
 - tbr_send, 74
 - tbr_t, 74
- tbr_connect
 - tbr.c, 72
 - tbr.h, 74
- tbr_disconnect
 - tbr.c, 72
 - tbr.h, 74
- tbr_recv
 - tbr.c, 73
 - tbr.h, 74
- tbr_reset
 - tbr.c, 73
 - tbr.h, 74
- tbr_send
 - tbr.c, 73
 - tbr.h, 74
- tbr_t
 - tbr.h, 74
- tempbuf
 - serial.c, 69
- tennis_ball, 15
 - _ball_image_hue_filter, 15
 - _init_particle_filter, 16
 - is_ball_middle, 16
 - particle_filter, 16
 - run, 16
- tennis_ball_runner, 16
- tennis_ball_test, 16
 - ballMissing, 16
 - ballPresent, 16
 - unitTest, 16
- test_box_contains
 - prquadtree_test::TestBox, 36
- test_box_insert
 - prquadtree_test::TestBox, 36
- test_insert
 - prquadtree_test::TestPrQuadTree, 38
- test_nearby
 - prquadtree_test::TestPrQuadTree, 38
- test_particle_insert
 - prquadtree_test::TestParticle, 37
- test_point_insert
 - prquadtree_test::TestPoint, 37
- text
 - lse_stt.c, 78
- theta
 - pose2d, 29
- time
 - arm.cpp, 61
 - claw.cpp, 62
 - wheels.cpp, 64
- turnleft
 - wheels.cpp, 64
- turnright
 - wheels.cpp, 64
- UP
 - controller, 22
- unitTest
 - basket_test, 11
 - tennis_ball_test, 16
- val
 - note, 24
- velocity
 - HBridgeMotor, 23
- view
 - simple_agent.cpp, 44
- visual.h
 - get_objects, 90
 - set_objects, 90
 - start_visual, 90
 - stop_visual, 90
- WHEEL_DEVID
 - tbr.c, 72
- WHEEL_LB1
 - wheels.cpp, 63

WHEEL_LB2
 wheels.cpp, 63
 WHEEL_LM1
 wheels.cpp, 63
 WHEEL_LM2
 wheels.cpp, 63
 WHEEL_LT1
 wheels.cpp, 63
 WHEEL_LT2
 wheels.cpp, 63
 WHEEL_RB1
 wheels.cpp, 63
 WHEEL_RB2
 wheels.cpp, 63
 WHEEL_RM1
 wheels.cpp, 63
 WHEEL_RM2
 wheels.cpp, 63
 WHEEL_RT1
 wheels.cpp, 64
 WHEEL_RT2
 wheels.cpp, 64
 wakeup
 agent.c, 41
 wheel_l
 wheels.cpp, 64
 wheel_r
 wheels.cpp, 64
 wheels.cpp
 backward, 64
 DEV_ID, 63
 forward, 64
 limit, 64
 loop, 64
 msg, 64
 setup, 64
 setwheels, 64
 stopwheels, 64
 time, 64
 turnleft, 64
 turnright, 64
 WHEEL_LB1, 63
 WHEEL_LB2, 63
 WHEEL_LM1, 63
 WHEEL_LM2, 63
 WHEEL_LT1, 63
 WHEEL_LT2, 63
 WHEEL_RB1, 63
 WHEEL_RB2, 63
 WHEEL_RM1, 63
 WHEEL_RM2, 63
 WHEEL_RT1, 64
 WHEEL_RT2, 64
 wheel_l, 64
 wheel_r, 64
 write
 HBridgeMotor, 23

X

 controller, 22
 x
 controller, 22
 point2d, 28
 point3d, 28
 pose2d, 29
 pose3d, 29
 prquadtree::Particle, 25
 prquadtree::Point, 28
 Y
 controller, 22
 y
 controller, 22
 point2d, 28
 point3d, 28
 pose2d, 29
 pose3d, 29
 prquadtree::Particle, 25
 prquadtree::Point, 28
 yaw
 pose3d, 29
 z
 point3d, 28
 pose3d, 29