

ohCaptain

V1.0

Generated by Doxygen 1.8.12

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 boost Namespace Reference	9
5.2 boost::units Namespace Reference	9
5.3 boost::units::constants Namespace Reference	9
5.3.1 Function Documentation	10
5.3.1.1 operator""_LD() [1/2]	10
5.3.1.2 operator""_LD() [2/2]	10
5.3.2 Variable Documentation	10
5.3.2.1 a0	10
5.3.2.2 alpha	11
5.3.2.3 atm	11
5.3.2.4 b	11
5.3.2.5 c	11

5.3.2.6	c1	11
5.3.2.7	c2	11
5.3.2.8	e	11
5.3.2.9	Eh	12
5.3.2.10	eps0	12
5.3.2.11	F	12
5.3.2.12	G	12
5.3.2.13	g	12
5.3.2.14	G0	12
5.3.2.15	h	12
5.3.2.16	hbar	12
5.3.2.17	k	13
5.3.2.18	kB	13
5.3.2.19	ke	13
5.3.2.20	KJ	13
5.3.2.21	IP	13
5.3.2.22	M_PI_LD	13
5.3.2.23	me	13
5.3.2.24	mp	13
5.3.2.25	mP	14
5.3.2.26	NA	14
5.3.2.27	R	14
5.3.2.28	re	14
5.3.2.29	Rinf	14
5.3.2.30	RK	14
5.3.2.31	sigma	14
5.3.2.32	tP	15
5.3.2.33	u0	15
5.3.2.34	uB	15
5.3.2.35	uN	15

5.3.2.36	Z0	15
5.4	boost::units::literals Namespace Reference	15
5.4.1	Function Documentation	43
5.4.1.1	operator"""" _A() [1/2]	43
5.4.1.2	operator"""" _A() [2/2]	43
5.4.1.3	operator"""" _aA() [1/2]	43
5.4.1.4	operator"""" _aA() [2/2]	43
5.4.1.5	operator"""" _aBq()	44
5.4.1.6	operator"""" _aBq()	44
5.4.1.7	operator"""" _aC()	44
5.4.1.8	operator"""" _aC()	44
5.4.1.9	operator"""" _acd()	44
5.4.1.10	operator"""" _acd()	44
5.4.1.11	operator"""" _aday()	44
5.4.1.12	operator"""" _aday()	45
5.4.1.13	operator"""" _adeg()	45
5.4.1.14	operator"""" _adeg()	45
5.4.1.15	operator"""" _adegC()	45
5.4.1.16	operator"""" _adegC()	45
5.4.1.17	operator"""" _aF()	45
5.4.1.18	operator"""" _aF()	45
5.4.1.19	operator"""" _ag()	46
5.4.1.20	operator"""" _ag()	46
5.4.1.21	operator"""" _aGy()	46
5.4.1.22	operator"""" _aGy()	46
5.4.1.23	operator"""" _aH()	46
5.4.1.24	operator"""" _aH()	46
5.4.1.25	operator"""" _ah()	46
5.4.1.26	operator"""" _ah()	47
5.4.1.27	operator"""" _aHz()	47

5.4.1.28 operator"" "" _aHz() [2/2]	47
5.4.1.29 operator"" "" _aJ() [1/2]	47
5.4.1.30 operator"" "" _aJ() [2/2]	47
5.4.1.31 operator"" "" _aK() [1/2]	47
5.4.1.32 operator"" "" _aK() [2/2]	47
5.4.1.33 operator"" "" _akat() [1/2]	48
5.4.1.34 operator"" "" _akat() [2/2]	48
5.4.1.35 operator"" "" _al() [1/2]	48
5.4.1.36 operator"" "" _al() [2/2]	48
5.4.1.37 operator"" "" _aL() [1/2]	48
5.4.1.38 operator"" "" _aL() [2/2]	48
5.4.1.39 operator"" "" _alm() [1/2]	48
5.4.1.40 operator"" "" _alm() [2/2]	49
5.4.1.41 operator"" "" _alx() [1/2]	49
5.4.1.42 operator"" "" _alx() [2/2]	49
5.4.1.43 operator"" "" _am() [1/2]	49
5.4.1.44 operator"" "" _am() [2/2]	49
5.4.1.45 operator"" "" _amin() [1/2]	49
5.4.1.46 operator"" "" _amin() [2/2]	49
5.4.1.47 operator"" "" _amol() [1/2]	50
5.4.1.48 operator"" "" _amol() [2/2]	50
5.4.1.49 operator"" "" _aN() [1/2]	50
5.4.1.50 operator"" "" _aN() [2/2]	50
5.4.1.51 operator"" "" _aohm() [1/2]	50
5.4.1.52 operator"" "" _aohm() [2/2]	50
5.4.1.53 operator"" "" _aPa() [1/2]	50
5.4.1.54 operator"" "" _aPa() [2/2]	51
5.4.1.55 operator"" "" _arad() [1/2]	51
5.4.1.56 operator"" "" _arad() [2/2]	51
5.4.1.57 operator"" "" _as() [1/2]	51

5.4.1.58 operator"" _as() [2/2]	51
5.4.1.59 operator"" _aS() [1/2]	51
5.4.1.60 operator"" _aS() [2/2]	51
5.4.1.61 operator"" _asr() [1/2]	52
5.4.1.62 operator"" _asr() [2/2]	52
5.4.1.63 operator"" _aSv() [1/2]	52
5.4.1.64 operator"" _aSv() [2/2]	52
5.4.1.65 operator"" _aT() [1/2]	52
5.4.1.66 operator"" _aT() [2/2]	52
5.4.1.67 operator"" _at() [1/2]	52
5.4.1.68 operator"" _at() [2/2]	53
5.4.1.69 operator"" _aV() [1/2]	53
5.4.1.70 operator"" _aV() [2/2]	53
5.4.1.71 operator"" _aW() [1/2]	53
5.4.1.72 operator"" _aW() [2/2]	53
5.4.1.73 operator"" _aWb() [1/2]	53
5.4.1.74 operator"" _aWb() [2/2]	53
5.4.1.75 operator"" _Bq() [1/2]	54
5.4.1.76 operator"" _Bq() [2/2]	54
5.4.1.77 operator"" _C() [1/2]	54
5.4.1.78 operator"" _C() [2/2]	54
5.4.1.79 operator"" _cA() [1/2]	54
5.4.1.80 operator"" _cA() [2/2]	54
5.4.1.81 operator"" _cBq() [1/2]	54
5.4.1.82 operator"" _cBq() [2/2]	55
5.4.1.83 operator"" _cC() [1/2]	55
5.4.1.84 operator"" _cC() [2/2]	55
5.4.1.85 operator"" _ccd() [1/2]	55
5.4.1.86 operator"" _ccd() [2/2]	55
5.4.1.87 operator"" _cd() [1/2]	55

5.4.1.88 operator"" _cd() [2/2]	55
5.4.1.89 operator"" _cday() [1/2]	56
5.4.1.90 operator"" _cday() [2/2]	56
5.4.1.91 operator"" _cdeg() [1/2]	56
5.4.1.92 operator"" _cdeg() [2/2]	56
5.4.1.93 operator"" _cdegC() [1/2]	56
5.4.1.94 operator"" _cdegC() [2/2]	56
5.4.1.95 operator"" _cF() [1/2]	56
5.4.1.96 operator"" _cF() [2/2]	57
5.4.1.97 operator"" _cg() [1/2]	57
5.4.1.98 operator"" _cg() [2/2]	57
5.4.1.99 operator"" _cGy() [1/2]	57
5.4.1.100 operator"" _cGy() [2/2]	57
5.4.1.101 operator"" _cH() [1/2]	57
5.4.1.102 operator"" _cH() [2/2]	57
5.4.1.103 operator"" _ch() [1/2]	58
5.4.1.104 operator"" _ch() [2/2]	58
5.4.1.105 operator"" _cHz() [1/2]	58
5.4.1.106 operator"" _cHz() [2/2]	58
5.4.1.107 operator"" _cJ() [1/2]	58
5.4.1.108 operator"" _cJ() [2/2]	58
5.4.1.109 operator"" _cK() [1/2]	58
5.4.1.110 operator"" _cK() [2/2]	59
5.4.1.111 operator"" _ckat() [1/2]	59
5.4.1.112 operator"" _ckat() [2/2]	59
5.4.1.113 operator"" _cl() [1/2]	59
5.4.1.114 operator"" _cl() [2/2]	59
5.4.1.115 operator"" _cL() [1/2]	59
5.4.1.116 operator"" _cL() [2/2]	59
5.4.1.117 operator"" _clm() [1/2]	60

5.4.1.118 operator"" "" _clm() [2/2]	60
5.4.1.119 operator"" "" _clk() [1/2]	60
5.4.1.120 operator"" "" _clk() [2/2]	60
5.4.1.121 operator"" "" _cm() [1/2]	60
5.4.1.122 operator"" "" _cm() [2/2]	60
5.4.1.123 operator"" "" _cmin() [1/2]	60
5.4.1.124 operator"" "" _cmin() [2/2]	61
5.4.1.125 operator"" "" _cmol() [1/2]	61
5.4.1.126 operator"" "" _cmol() [2/2]	61
5.4.1.127 operator"" "" _cN() [1/2]	61
5.4.1.128 operator"" "" _cN() [2/2]	61
5.4.1.129 operator"" "" _cohdm() [1/2]	61
5.4.1.130 operator"" "" _cohdm() [2/2]	61
5.4.1.131 operator"" "" _cPa() [1/2]	62
5.4.1.132 operator"" "" _cPa() [2/2]	62
5.4.1.133 operator"" "" _crad() [1/2]	62
5.4.1.134 operator"" "" _crad() [2/2]	62
5.4.1.135 operator"" "" _cs() [1/2]	62
5.4.1.136 operator"" "" _cs() [2/2]	62
5.4.1.137 operator"" "" _cS() [1/2]	62
5.4.1.138 operator"" "" _cS() [2/2]	63
5.4.1.139 operator"" "" _csr() [1/2]	63
5.4.1.140 operator"" "" _csr() [2/2]	63
5.4.1.141 operator"" "" _cSv() [1/2]	63
5.4.1.142 operator"" "" _cSv() [2/2]	63
5.4.1.143 operator"" "" _cT() [1/2]	63
5.4.1.144 operator"" "" _cT() [2/2]	63
5.4.1.145 operator"" "" _ct() [1/2]	64
5.4.1.146 operator"" "" _ct() [2/2]	64
5.4.1.147 operator"" "" _cV() [1/2]	64

5.4.1.148 operator"" "" _cV() [2/2]	64
5.4.1.149 operator"" "" _cW() [1/2]	64
5.4.1.150 operator"" "" _cW() [2/2]	64
5.4.1.151 operator"" "" _cWb() [1/2]	64
5.4.1.152 operator"" "" _cWb() [2/2]	65
5.4.1.153 operator"" "" _dA() [1/2]	65
5.4.1.154 operator"" "" _dA() [2/2]	65
5.4.1.155 operator"" "" _daA() [1/2]	65
5.4.1.156 operator"" "" _daA() [2/2]	65
5.4.1.157 operator"" "" _daBq() [1/2]	65
5.4.1.158 operator"" "" _daBq() [2/2]	65
5.4.1.159 operator"" "" _daC() [1/2]	66
5.4.1.160 operator"" "" _daC() [2/2]	66
5.4.1.161 operator"" "" _dacd() [1/2]	66
5.4.1.162 operator"" "" _dacd() [2/2]	66
5.4.1.163 operator"" "" _daday() [1/2]	66
5.4.1.164 operator"" "" _daday() [2/2]	66
5.4.1.165 operator"" "" _dadeg() [1/2]	66
5.4.1.166 operator"" "" _dadeg() [2/2]	67
5.4.1.167 operator"" "" _dadegC() [1/2]	67
5.4.1.168 operator"" "" _dadegC() [2/2]	67
5.4.1.169 operator"" "" _daF() [1/2]	67
5.4.1.170 operator"" "" _daF() [2/2]	67
5.4.1.171 operator"" "" _dag() [1/2]	67
5.4.1.172 operator"" "" _dag() [2/2]	67
5.4.1.173 operator"" "" _daGy() [1/2]	68
5.4.1.174 operator"" "" _daGy() [2/2]	68
5.4.1.175 operator"" "" _daH() [1/2]	68
5.4.1.176 operator"" "" _daH() [2/2]	68
5.4.1.177 operator"" "" _dah() [1/2]	68

5.4.1.178 operator"" "" _dah() [2/2]	68
5.4.1.179 operator"" "" _daHz() [1/2]	68
5.4.1.180 operator"" "" _daHz() [2/2]	69
5.4.1.181 operator"" "" _daJ() [1/2]	69
5.4.1.182 operator"" "" _daJ() [2/2]	69
5.4.1.183 operator"" "" _daK() [1/2]	69
5.4.1.184 operator"" "" _daK() [2/2]	69
5.4.1.185 operator"" "" _dakat() [1/2]	69
5.4.1.186 operator"" "" _dakat() [2/2]	69
5.4.1.187 operator"" "" _dal() [1/2]	70
5.4.1.188 operator"" "" _dal() [2/2]	70
5.4.1.189 operator"" "" _daL() [1/2]	70
5.4.1.190 operator"" "" _daL() [2/2]	70
5.4.1.191 operator"" "" _dalm() [1/2]	70
5.4.1.192 operator"" "" _dalm() [2/2]	70
5.4.1.193 operator"" "" _dalx() [1/2]	70
5.4.1.194 operator"" "" _dalx() [2/2]	71
5.4.1.195 operator"" "" _dam() [1/2]	71
5.4.1.196 operator"" "" _dam() [2/2]	71
5.4.1.197 operator"" "" _damin() [1/2]	71
5.4.1.198 operator"" "" _damin() [2/2]	71
5.4.1.199 operator"" "" _damol() [1/2]	71
5.4.1.200 operator"" "" _damol() [2/2]	71
5.4.1.201 operator"" "" _daN() [1/2]	72
5.4.1.202 operator"" "" _daN() [2/2]	72
5.4.1.203 operator"" "" _daohm() [1/2]	72
5.4.1.204 operator"" "" _daohm() [2/2]	72
5.4.1.205 operator"" "" _daPa() [1/2]	72
5.4.1.206 operator"" "" _daPa() [2/2]	72
5.4.1.207 operator"" "" _darad() [1/2]	72

5.4.1.208 operator"" _darad() [2/2]	73
5.4.1.209 operator"" _das() [1/2]	73
5.4.1.210 operator"" _das() [2/2]	73
5.4.1.211 operator"" _daS() [1/2]	73
5.4.1.212 operator"" _daS() [2/2]	73
5.4.1.213 operator"" _dasr() [1/2]	73
5.4.1.214 operator"" _dasr() [2/2]	73
5.4.1.215 operator"" _daSv() [1/2]	74
5.4.1.216 operator"" _daSv() [2/2]	74
5.4.1.217 operator"" _daT() [1/2]	74
5.4.1.218 operator"" _daT() [2/2]	74
5.4.1.219 operator"" _dat() [1/2]	74
5.4.1.220 operator"" _dat() [2/2]	74
5.4.1.221 operator"" _daV() [1/2]	74
5.4.1.222 operator"" _daV() [2/2]	75
5.4.1.223 operator"" _daW() [1/2]	75
5.4.1.224 operator"" _daW() [2/2]	75
5.4.1.225 operator"" _daWb() [1/2]	75
5.4.1.226 operator"" _daWb() [2/2]	75
5.4.1.227 operator"" _day() [1/2]	75
5.4.1.228 operator"" _day() [2/2]	75
5.4.1.229 operator"" _dBq() [1/2]	76
5.4.1.230 operator"" _dBq() [2/2]	76
5.4.1.231 operator"" _dC() [1/2]	76
5.4.1.232 operator"" _dC() [2/2]	76
5.4.1.233 operator"" _dcdf() [1/2]	76
5.4.1.234 operator"" _dcdf() [2/2]	76
5.4.1.235 operator"" _dday() [1/2]	76
5.4.1.236 operator"" _dday() [2/2]	77
5.4.1.237 operator"" _ddeg() [1/2]	77

5.4.1.238 operator"" _ddeg() [2/2]	77
5.4.1.239 operator"" _ddegC() [1/2]	77
5.4.1.240 operator"" _ddegC() [2/2]	77
5.4.1.241 operator"" _deg() [1/2]	77
5.4.1.242 operator"" _deg() [2/2]	77
5.4.1.243 operator"" _degC() [1/2]	78
5.4.1.244 operator"" _degC() [2/2]	78
5.4.1.245 operator"" _dF() [1/2]	78
5.4.1.246 operator"" _dF() [2/2]	78
5.4.1.247 operator"" _dg() [1/2]	78
5.4.1.248 operator"" _dg() [2/2]	78
5.4.1.249 operator"" _dGy() [1/2]	78
5.4.1.250 operator"" _dGy() [2/2]	79
5.4.1.251 operator"" _dH() [1/2]	79
5.4.1.252 operator"" _dH() [2/2]	79
5.4.1.253 operator"" _dh() [1/2]	79
5.4.1.254 operator"" _dh() [2/2]	79
5.4.1.255 operator"" _dHz() [1/2]	79
5.4.1.256 operator"" _dHz() [2/2]	79
5.4.1.257 operator"" _dJ() [1/2]	80
5.4.1.258 operator"" _dJ() [2/2]	80
5.4.1.259 operator"" _dK() [1/2]	80
5.4.1.260 operator"" _dK() [2/2]	80
5.4.1.261 operator"" _dkat() [1/2]	80
5.4.1.262 operator"" _dkat() [2/2]	80
5.4.1.263 operator"" _dl() [1/2]	80
5.4.1.264 operator"" _dl() [2/2]	81
5.4.1.265 operator"" _dL() [1/2]	81
5.4.1.266 operator"" _dL() [2/2]	81
5.4.1.267 operator"" _dlm() [1/2]	81

5.4.1.268 operator"" "" _dlm() [2/2]	81
5.4.1.269 operator"" "" _dlx() [1/2]	81
5.4.1.270 operator"" "" _dlx() [2/2]	81
5.4.1.271 operator"" "" _dm() [1/2]	82
5.4.1.272 operator"" "" _dm() [2/2]	82
5.4.1.273 operator"" "" _dmin() [1/2]	82
5.4.1.274 operator"" "" _dmin() [2/2]	82
5.4.1.275 operator"" "" _dmol() [1/2]	82
5.4.1.276 operator"" "" _dmol() [2/2]	82
5.4.1.277 operator"" "" _dN() [1/2]	82
5.4.1.278 operator"" "" _dN() [2/2]	83
5.4.1.279 operator"" "" _dohm() [1/2]	83
5.4.1.280 operator"" "" _dohm() [2/2]	83
5.4.1.281 operator"" "" _dPa() [1/2]	83
5.4.1.282 operator"" "" _dPa() [2/2]	83
5.4.1.283 operator"" "" _drad() [1/2]	83
5.4.1.284 operator"" "" _drad() [2/2]	83
5.4.1.285 operator"" "" _ds() [1/2]	84
5.4.1.286 operator"" "" _ds() [2/2]	84
5.4.1.287 operator"" "" _dS() [1/2]	84
5.4.1.288 operator"" "" _dS() [2/2]	84
5.4.1.289 operator"" "" _dsr() [1/2]	84
5.4.1.290 operator"" "" _dsr() [2/2]	84
5.4.1.291 operator"" "" _dSv() [1/2]	84
5.4.1.292 operator"" "" _dSv() [2/2]	85
5.4.1.293 operator"" "" _dT() [1/2]	85
5.4.1.294 operator"" "" _dT() [2/2]	85
5.4.1.295 operator"" "" _dt() [1/2]	85
5.4.1.296 operator"" "" _dt() [2/2]	85
5.4.1.297 operator"" "" _dV() [1/2]	85

5.4.1.298 operator"" "" _dV() [2/2]	85
5.4.1.299 operator"" "" _dW() [1/2]	86
5.4.1.300 operator"" "" _dW() [2/2]	86
5.4.1.301 operator"" "" _dWb() [1/2]	86
5.4.1.302 operator"" "" _dWb() [2/2]	86
5.4.1.303 operator"" "" _EA() [1/2]	86
5.4.1.304 operator"" "" _EA() [2/2]	86
5.4.1.305 operator"" "" _EBq() [1/2]	86
5.4.1.306 operator"" "" _EBq() [2/2]	87
5.4.1.307 operator"" "" _EC() [1/2]	87
5.4.1.308 operator"" "" _EC() [2/2]	87
5.4.1.309 operator"" "" _Ecd() [1/2]	87
5.4.1.310 operator"" "" _Ecd() [2/2]	87
5.4.1.311 operator"" "" _Eday() [1/2]	87
5.4.1.312 operator"" "" _Eday() [2/2]	87
5.4.1.313 operator"" "" _Edeg() [1/2]	88
5.4.1.314 operator"" "" _Edeg() [2/2]	88
5.4.1.315 operator"" "" _EdegC() [1/2]	88
5.4.1.316 operator"" "" _EdegC() [2/2]	88
5.4.1.317 operator"" "" _EF() [1/2]	88
5.4.1.318 operator"" "" _EF() [2/2]	88
5.4.1.319 operator"" "" _Eg() [1/2]	88
5.4.1.320 operator"" "" _Eg() [2/2]	89
5.4.1.321 operator"" "" _EGy() [1/2]	89
5.4.1.322 operator"" "" _EGy() [2/2]	89
5.4.1.323 operator"" "" _EH() [1/2]	89
5.4.1.324 operator"" "" _EH() [2/2]	89
5.4.1.325 operator"" "" _Eh() [1/2]	89
5.4.1.326 operator"" "" _Eh() [2/2]	89
5.4.1.327 operator"" "" _EHz() [1/2]	90

5.4.1.328 operator"" _EHz() [2/2]	90
5.4.1.329 operator"" _EJ() [1/2]	90
5.4.1.330 operator"" _EJ() [2/2]	90
5.4.1.331 operator"" _EK() [1/2]	90
5.4.1.332 operator"" _EK() [2/2]	90
5.4.1.333 operator"" _Ekat() [1/2]	90
5.4.1.334 operator"" _Ekat() [2/2]	91
5.4.1.335 operator"" _El() [1/2]	91
5.4.1.336 operator"" _El() [2/2]	91
5.4.1.337 operator"" _EL() [1/2]	91
5.4.1.338 operator"" _EL() [2/2]	91
5.4.1.339 operator"" _Elm() [1/2]	91
5.4.1.340 operator"" _Elm() [2/2]	91
5.4.1.341 operator"" _Elx() [1/2]	92
5.4.1.342 operator"" _Elx() [2/2]	92
5.4.1.343 operator"" _Em() [1/2]	92
5.4.1.344 operator"" _Em() [2/2]	92
5.4.1.345 operator"" _Emin() [1/2]	92
5.4.1.346 operator"" _Emin() [2/2]	92
5.4.1.347 operator"" _Emol() [1/2]	92
5.4.1.348 operator"" _Emol() [2/2]	93
5.4.1.349 operator"" _EN() [1/2]	93
5.4.1.350 operator"" _EN() [2/2]	93
5.4.1.351 operator"" _Eohm() [1/2]	93
5.4.1.352 operator"" _Eohm() [2/2]	93
5.4.1.353 operator"" _EPa() [1/2]	93
5.4.1.354 operator"" _EPa() [2/2]	93
5.4.1.355 operator"" _Erad() [1/2]	94
5.4.1.356 operator"" _Erad() [2/2]	94
5.4.1.357 operator"" _Es() [1/2]	94

5.4.1.358 operator"" "" _Es() [2/2]	94
5.4.1.359 operator"" "" _ES() [1/2]	94
5.4.1.360 operator"" "" _ES() [2/2]	94
5.4.1.361 operator"" "" _Esr() [1/2]	94
5.4.1.362 operator"" "" _Esr() [2/2]	95
5.4.1.363 operator"" "" _ESv() [1/2]	95
5.4.1.364 operator"" "" _ESv() [2/2]	95
5.4.1.365 operator"" "" _ET() [1/2]	95
5.4.1.366 operator"" "" _ET() [2/2]	95
5.4.1.367 operator"" "" _Et() [1/2]	95
5.4.1.368 operator"" "" _Et() [2/2]	95
5.4.1.369 operator"" "" _EV() [1/2]	96
5.4.1.370 operator"" "" _EV() [2/2]	96
5.4.1.371 operator"" "" _EW() [1/2]	96
5.4.1.372 operator"" "" _EW() [2/2]	96
5.4.1.373 operator"" "" _EWb() [1/2]	96
5.4.1.374 operator"" "" _EWb() [2/2]	96
5.4.1.375 operator"" "" _F() [1/2]	96
5.4.1.376 operator"" "" _F() [2/2]	97
5.4.1.377 operator"" "" _fA() [1/2]	97
5.4.1.378 operator"" "" _fA() [2/2]	97
5.4.1.379 operator"" "" _fBq() [1/2]	97
5.4.1.380 operator"" "" _fBq() [2/2]	97
5.4.1.381 operator"" "" _fC() [1/2]	97
5.4.1.382 operator"" "" _fC() [2/2]	97
5.4.1.383 operator"" "" _fcd() [1/2]	98
5.4.1.384 operator"" "" _fcd() [2/2]	98
5.4.1.385 operator"" "" _fday() [1/2]	98
5.4.1.386 operator"" "" _fday() [2/2]	98
5.4.1.387 operator"" "" _fdeg() [1/2]	98

5.4.1.388 operator"" _fdeg() [2/2]	98
5.4.1.389 operator"" _fdegC() [1/2]	98
5.4.1.390 operator"" _fdegC() [2/2]	99
5.4.1.391 operator"" _fF() [1/2]	99
5.4.1.392 operator"" _fF() [2/2]	99
5.4.1.393 operator"" _fg() [1/2]	99
5.4.1.394 operator"" _fg() [2/2]	99
5.4.1.395 operator"" _fGy() [1/2]	99
5.4.1.396 operator"" _fGy() [2/2]	99
5.4.1.397 operator"" _fH() [1/2]	100
5.4.1.398 operator"" _fH() [2/2]	100
5.4.1.399 operator"" _fh() [1/2]	100
5.4.1.400 operator"" _fh() [2/2]	100
5.4.1.401 operator"" _fHz() [1/2]	100
5.4.1.402 operator"" _fHz() [2/2]	100
5.4.1.403 operator"" _fJ() [1/2]	100
5.4.1.404 operator"" _fJ() [2/2]	101
5.4.1.405 operator"" _fK() [1/2]	101
5.4.1.406 operator"" _fK() [2/2]	101
5.4.1.407 operator"" _fkat() [1/2]	101
5.4.1.408 operator"" _fkat() [2/2]	101
5.4.1.409 operator"" _fl() [1/2]	101
5.4.1.410 operator"" _fl() [2/2]	101
5.4.1.411 operator"" _fL() [1/2]	102
5.4.1.412 operator"" _fL() [2/2]	102
5.4.1.413 operator"" _flm() [1/2]	102
5.4.1.414 operator"" _flm() [2/2]	102
5.4.1.415 operator"" _flx() [1/2]	102
5.4.1.416 operator"" _flx() [2/2]	102
5.4.1.417 operator"" _fm() [1/2]	102

5.4.1.418 operator"" _fm() [2/2]	103
5.4.1.419 operator"" _fmin() [1/2]	103
5.4.1.420 operator"" _fmin() [2/2]	103
5.4.1.421 operator"" _fmol() [1/2]	103
5.4.1.422 operator"" _fmol() [2/2]	103
5.4.1.423 operator"" _fN() [1/2]	103
5.4.1.424 operator"" _fN() [2/2]	103
5.4.1.425 operator"" _fohm() [1/2]	104
5.4.1.426 operator"" _fohm() [2/2]	104
5.4.1.427 operator"" _fPa() [1/2]	104
5.4.1.428 operator"" _fPa() [2/2]	104
5.4.1.429 operator"" _frad() [1/2]	104
5.4.1.430 operator"" _frad() [2/2]	104
5.4.1.431 operator"" _fs() [1/2]	104
5.4.1.432 operator"" _fs() [2/2]	105
5.4.1.433 operator"" _fS() [1/2]	105
5.4.1.434 operator"" _fS() [2/2]	105
5.4.1.435 operator"" _fsr() [1/2]	105
5.4.1.436 operator"" _fsr() [2/2]	105
5.4.1.437 operator"" _fSv() [1/2]	105
5.4.1.438 operator"" _fSv() [2/2]	105
5.4.1.439 operator"" _fT() [1/2]	106
5.4.1.440 operator"" _fT() [2/2]	106
5.4.1.441 operator"" _ft() [1/2]	106
5.4.1.442 operator"" _ft() [2/2]	106
5.4.1.443 operator"" _fV() [1/2]	106
5.4.1.444 operator"" _fV() [2/2]	106
5.4.1.445 operator"" _fW() [1/2]	106
5.4.1.446 operator"" _fW() [2/2]	107
5.4.1.447 operator"" _fWb() [1/2]	107

5.4.1.448 operator"" "" _fWb() [2/2]	107
5.4.1.449 operator"" "" _g() [1/2]	107
5.4.1.450 operator"" "" _g() [2/2]	107
5.4.1.451 operator"" "" _GA() [1/2]	107
5.4.1.452 operator"" "" _GA() [2/2]	107
5.4.1.453 operator"" "" _GBq() [1/2]	108
5.4.1.454 operator"" "" _GBq() [2/2]	108
5.4.1.455 operator"" "" _GC() [1/2]	108
5.4.1.456 operator"" "" _GC() [2/2]	108
5.4.1.457 operator"" "" _Gcd() [1/2]	108
5.4.1.458 operator"" "" _Gcd() [2/2]	108
5.4.1.459 operator"" "" _Gday() [1/2]	108
5.4.1.460 operator"" "" _Gday() [2/2]	109
5.4.1.461 operator"" "" _Gdeg() [1/2]	109
5.4.1.462 operator"" "" _Gdeg() [2/2]	109
5.4.1.463 operator"" "" _GdegC() [1/2]	109
5.4.1.464 operator"" "" _GdegC() [2/2]	109
5.4.1.465 operator"" "" _GF() [1/2]	109
5.4.1.466 operator"" "" _GF() [2/2]	109
5.4.1.467 operator"" "" _Gg() [1/2]	110
5.4.1.468 operator"" "" _Gg() [2/2]	110
5.4.1.469 operator"" "" _GGy() [1/2]	110
5.4.1.470 operator"" "" _GGy() [2/2]	110
5.4.1.471 operator"" "" _GH() [1/2]	110
5.4.1.472 operator"" "" _GH() [2/2]	110
5.4.1.473 operator"" "" _Gh() [1/2]	110
5.4.1.474 operator"" "" _Gh() [2/2]	111
5.4.1.475 operator"" "" _GHz() [1/2]	111
5.4.1.476 operator"" "" _GHz() [2/2]	111
5.4.1.477 operator"" "" _GJ() [1/2]	111

5.4.1.478 operator"" _GJ() [2/2]	111
5.4.1.479 operator"" _GK() [1/2]	111
5.4.1.480 operator"" _GK() [2/2]	111
5.4.1.481 operator"" _Gkat() [1/2]	112
5.4.1.482 operator"" _Gkat() [2/2]	112
5.4.1.483 operator"" _Gl() [1/2]	112
5.4.1.484 operator"" _Gl() [2/2]	112
5.4.1.485 operator"" _GL() [1/2]	112
5.4.1.486 operator"" _GL() [2/2]	112
5.4.1.487 operator"" _Glm() [1/2]	112
5.4.1.488 operator"" _Glm() [2/2]	113
5.4.1.489 operator"" _Glx() [1/2]	113
5.4.1.490 operator"" _Glx() [2/2]	113
5.4.1.491 operator"" _Gm() [1/2]	113
5.4.1.492 operator"" _Gm() [2/2]	113
5.4.1.493 operator"" _Gmin() [1/2]	113
5.4.1.494 operator"" _Gmin() [2/2]	113
5.4.1.495 operator"" _Gmol() [1/2]	114
5.4.1.496 operator"" _Gmol() [2/2]	114
5.4.1.497 operator"" _GN() [1/2]	114
5.4.1.498 operator"" _GN() [2/2]	114
5.4.1.499 operator"" _Gohm() [1/2]	114
5.4.1.500 operator"" _Gohm() [2/2]	114
5.4.1.501 operator"" _GPa() [1/2]	114
5.4.1.502 operator"" _GPa() [2/2]	115
5.4.1.503 operator"" _Grad() [1/2]	115
5.4.1.504 operator"" _Grad() [2/2]	115
5.4.1.505 operator"" _Gs() [1/2]	115
5.4.1.506 operator"" _Gs() [2/2]	115
5.4.1.507 operator"" _GS() [1/2]	115

5.4.1.508 operator"" _GS() [2/2]	115
5.4.1.509 operator"" _Gsr() [1/2]	116
5.4.1.510 operator"" _Gsr() [2/2]	116
5.4.1.511 operator"" _GSv() [1/2]	116
5.4.1.512 operator"" _GSv() [2/2]	116
5.4.1.513 operator"" _GT() [1/2]	116
5.4.1.514 operator"" _GT() [2/2]	116
5.4.1.515 operator"" _Gt() [1/2]	116
5.4.1.516 operator"" _Gt() [2/2]	117
5.4.1.517 operator"" _GV() [1/2]	117
5.4.1.518 operator"" _GV() [2/2]	117
5.4.1.519 operator"" _GW() [1/2]	117
5.4.1.520 operator"" _GW() [2/2]	117
5.4.1.521 operator"" _Gwb() [1/2]	117
5.4.1.522 operator"" _Gwb() [2/2]	117
5.4.1.523 operator"" _Gy() [1/2]	118
5.4.1.524 operator"" _Gy() [2/2]	118
5.4.1.525 operator"" _H() [1/2]	118
5.4.1.526 operator"" _H() [2/2]	118
5.4.1.527 operator"" _h() [1/2]	118
5.4.1.528 operator"" _h() [2/2]	118
5.4.1.529 operator"" _hA() [1/2]	118
5.4.1.530 operator"" _hA() [2/2]	119
5.4.1.531 operator"" _hBq() [1/2]	119
5.4.1.532 operator"" _hBq() [2/2]	119
5.4.1.533 operator"" _hC() [1/2]	119
5.4.1.534 operator"" _hC() [2/2]	119
5.4.1.535 operator"" _hcd() [1/2]	119
5.4.1.536 operator"" _hcd() [2/2]	119
5.4.1.537 operator"" _hday() [1/2]	120

5.4.1.538 operator"" _hday() [2/2]	120
5.4.1.539 operator"" _hdeg() [1/2]	120
5.4.1.540 operator"" _hdeg() [2/2]	120
5.4.1.541 operator"" _hdegC() [1/2]	120
5.4.1.542 operator"" _hdegC() [2/2]	120
5.4.1.543 operator"" _hF() [1/2]	120
5.4.1.544 operator"" _hF() [2/2]	121
5.4.1.545 operator"" _hg() [1/2]	121
5.4.1.546 operator"" _hg() [2/2]	121
5.4.1.547 operator"" _hGy() [1/2]	121
5.4.1.548 operator"" _hGy() [2/2]	121
5.4.1.549 operator"" _hH() [1/2]	121
5.4.1.550 operator"" _hH() [2/2]	121
5.4.1.551 operator"" _hh() [1/2]	122
5.4.1.552 operator"" _hh() [2/2]	122
5.4.1.553 operator"" _hHz() [1/2]	122
5.4.1.554 operator"" _hHz() [2/2]	122
5.4.1.555 operator"" _hJ() [1/2]	122
5.4.1.556 operator"" _hJ() [2/2]	122
5.4.1.557 operator"" _hK() [1/2]	122
5.4.1.558 operator"" _hK() [2/2]	123
5.4.1.559 operator"" _hkat() [1/2]	123
5.4.1.560 operator"" _hkat() [2/2]	123
5.4.1.561 operator"" _hl() [1/2]	123
5.4.1.562 operator"" _hl() [2/2]	123
5.4.1.563 operator"" _hL() [1/2]	123
5.4.1.564 operator"" _hL() [2/2]	123
5.4.1.565 operator"" _hlm() [1/2]	124
5.4.1.566 operator"" _hlm() [2/2]	124
5.4.1.567 operator"" _hlx() [1/2]	124

5.4.1.568 operator"" _hlx() [2/2]	124
5.4.1.569 operator"" _hm() [1/2]	124
5.4.1.570 operator"" _hm() [2/2]	124
5.4.1.571 operator"" _hmin() [1/2]	124
5.4.1.572 operator"" _hmin() [2/2]	125
5.4.1.573 operator"" _hmo() [1/2]	125
5.4.1.574 operator"" _hmo() [2/2]	125
5.4.1.575 operator"" _hN() [1/2]	125
5.4.1.576 operator"" _hN() [2/2]	125
5.4.1.577 operator"" _hohm() [1/2]	125
5.4.1.578 operator"" _hohm() [2/2]	125
5.4.1.579 operator"" _hPa() [1/2]	126
5.4.1.580 operator"" _hPa() [2/2]	126
5.4.1.581 operator"" _hrad() [1/2]	126
5.4.1.582 operator"" _hrad() [2/2]	126
5.4.1.583 operator"" _hs() [1/2]	126
5.4.1.584 operator"" _hs() [2/2]	126
5.4.1.585 operator"" _hS() [1/2]	126
5.4.1.586 operator"" _hS() [2/2]	127
5.4.1.587 operator"" _hsr() [1/2]	127
5.4.1.588 operator"" _hsr() [2/2]	127
5.4.1.589 operator"" _hSv() [1/2]	127
5.4.1.590 operator"" _hSv() [2/2]	127
5.4.1.591 operator"" _hT() [1/2]	127
5.4.1.592 operator"" _hT() [2/2]	127
5.4.1.593 operator"" _ht() [1/2]	128
5.4.1.594 operator"" _ht() [2/2]	128
5.4.1.595 operator"" _hV() [1/2]	128
5.4.1.596 operator"" _hV() [2/2]	128
5.4.1.597 operator"" _hW() [1/2]	128

5.4.1.598 operator"" _hW() [2/2]	128
5.4.1.599 operator"" _hWb() [1/2]	128
5.4.1.600 operator"" _hWb() [2/2]	129
5.4.1.601 operator"" _Hz() [1/2]	129
5.4.1.602 operator"" _Hz() [2/2]	129
5.4.1.603 operator"" _J() [1/2]	129
5.4.1.604 operator"" _J() [2/2]	129
5.4.1.605 operator"" _K() [1/2]	129
5.4.1.606 operator"" _K() [2/2]	129
5.4.1.607 operator"" _kA() [1/2]	130
5.4.1.608 operator"" _kA() [2/2]	130
5.4.1.609 operator"" _kat() [1/2]	130
5.4.1.610 operator"" _kat() [2/2]	130
5.4.1.611 operator"" _kBq() [1/2]	130
5.4.1.612 operator"" _kBq() [2/2]	130
5.4.1.613 operator"" _kC() [1/2]	130
5.4.1.614 operator"" _kC() [2/2]	131
5.4.1.615 operator"" _kcd() [1/2]	131
5.4.1.616 operator"" _kcd() [2/2]	131
5.4.1.617 operator"" _kday() [1/2]	131
5.4.1.618 operator"" _kday() [2/2]	131
5.4.1.619 operator"" _kdeg() [1/2]	131
5.4.1.620 operator"" _kdeg() [2/2]	131
5.4.1.621 operator"" _kdegC() [1/2]	132
5.4.1.622 operator"" _kdegC() [2/2]	132
5.4.1.623 operator"" _kF() [1/2]	132
5.4.1.624 operator"" _kF() [2/2]	132
5.4.1.625 operator"" _kg() [1/2]	132
5.4.1.626 operator"" _kg() [2/2]	132
5.4.1.627 operator"" _kGy() [1/2]	132

5.4.1.628 operator"" "" _kGy() [2/2]	133
5.4.1.629 operator"" "" _kH() [1/2]	133
5.4.1.630 operator"" "" _kH() [2/2]	133
5.4.1.631 operator"" "" _kh() [1/2]	133
5.4.1.632 operator"" "" _kh() [2/2]	133
5.4.1.633 operator"" "" _kHz() [1/2]	133
5.4.1.634 operator"" "" _kHz() [2/2]	133
5.4.1.635 operator"" "" _kJ() [1/2]	134
5.4.1.636 operator"" "" _kJ() [2/2]	134
5.4.1.637 operator"" "" _kK() [1/2]	134
5.4.1.638 operator"" "" _kK() [2/2]	134
5.4.1.639 operator"" "" _kkat() [1/2]	134
5.4.1.640 operator"" "" _kkat() [2/2]	134
5.4.1.641 operator"" "" _kl() [1/2]	134
5.4.1.642 operator"" "" _kl() [2/2]	135
5.4.1.643 operator"" "" _kL() [1/2]	135
5.4.1.644 operator"" "" _kL() [2/2]	135
5.4.1.645 operator"" "" _klm() [1/2]	135
5.4.1.646 operator"" "" _klm() [2/2]	135
5.4.1.647 operator"" "" _klx() [1/2]	135
5.4.1.648 operator"" "" _klx() [2/2]	135
5.4.1.649 operator"" "" _km() [1/2]	136
5.4.1.650 operator"" "" _km() [2/2]	136
5.4.1.651 operator"" "" _kmin() [1/2]	136
5.4.1.652 operator"" "" _kmin() [2/2]	136
5.4.1.653 operator"" "" _kmol() [1/2]	136
5.4.1.654 operator"" "" _kmol() [2/2]	136
5.4.1.655 operator"" "" _kN() [1/2]	136
5.4.1.656 operator"" "" _kN() [2/2]	137
5.4.1.657 operator"" "" _kohm() [1/2]	137

5.4.1.658 operator"" "" _kohm() [2/2]	137
5.4.1.659 operator"" "" _kPa() [1/2]	137
5.4.1.660 operator"" "" _kPa() [2/2]	137
5.4.1.661 operator"" "" _krad() [1/2]	137
5.4.1.662 operator"" "" _krad() [2/2]	137
5.4.1.663 operator"" "" _ks() [1/2]	138
5.4.1.664 operator"" "" _ks() [2/2]	138
5.4.1.665 operator"" "" _kS() [1/2]	138
5.4.1.666 operator"" "" _kS() [2/2]	138
5.4.1.667 operator"" "" _ksr() [1/2]	138
5.4.1.668 operator"" "" _ksr() [2/2]	138
5.4.1.669 operator"" "" _kSv() [1/2]	138
5.4.1.670 operator"" "" _kSv() [2/2]	139
5.4.1.671 operator"" "" _kT() [1/2]	139
5.4.1.672 operator"" "" _kT() [2/2]	139
5.4.1.673 operator"" "" _kt() [1/2]	139
5.4.1.674 operator"" "" _kt() [2/2]	139
5.4.1.675 operator"" "" _kV() [1/2]	139
5.4.1.676 operator"" "" _kV() [2/2]	139
5.4.1.677 operator"" "" _kW() [1/2]	140
5.4.1.678 operator"" "" _kW() [2/2]	140
5.4.1.679 operator"" "" _kWb() [1/2]	140
5.4.1.680 operator"" "" _kWb() [2/2]	140
5.4.1.681 operator"" "" _l() [1/2]	140
5.4.1.682 operator"" "" _l() [2/2]	140
5.4.1.683 operator"" "" _L() [1/2]	140
5.4.1.684 operator"" "" _L() [2/2]	141
5.4.1.685 operator"" "" _Im() [1/2]	141
5.4.1.686 operator"" "" _Im() [2/2]	141
5.4.1.687 operator"" "" _Ix() [1/2]	141

5.4.1.688 operator"" "" _lx() [2/2]	141
5.4.1.689 operator"" "" _m() [1/2]	141
5.4.1.690 operator"" "" _m() [2/2]	141
5.4.1.691 operator"" "" _MA() [1/2]	142
5.4.1.692 operator"" "" _mA() [1/2]	142
5.4.1.693 operator"" "" _mA() [2/2]	142
5.4.1.694 operator"" "" _MA() [2/2]	142
5.4.1.695 operator"" "" _MBq() [1/2]	142
5.4.1.696 operator"" "" _MBq() [2/2]	142
5.4.1.697 operator"" "" _mBq() [1/2]	142
5.4.1.698 operator"" "" _mBq() [2/2]	143
5.4.1.699 operator"" "" _mC() [1/2]	143
5.4.1.700 operator"" "" _MC() [1/2]	143
5.4.1.701 operator"" "" _MC() [2/2]	143
5.4.1.702 operator"" "" _mC() [2/2]	143
5.4.1.703 operator"" "" _Mcd() [1/2]	143
5.4.1.704 operator"" "" _Mcd() [2/2]	143
5.4.1.705 operator"" "" _mcd() [1/2]	144
5.4.1.706 operator"" "" _mcd() [2/2]	144
5.4.1.707 operator"" "" _Mday() [1/2]	144
5.4.1.708 operator"" "" _mday() [1/2]	144
5.4.1.709 operator"" "" _mday() [2/2]	144
5.4.1.710 operator"" "" _Mday() [2/2]	144
5.4.1.711 operator"" "" _Mdeg() [1/2]	144
5.4.1.712 operator"" "" _Mdeg() [2/2]	145
5.4.1.713 operator"" "" _mdeg() [1/2]	145
5.4.1.714 operator"" "" _mdeg() [2/2]	145
5.4.1.715 operator"" "" _MdegC() [1/2]	145
5.4.1.716 operator"" "" _MdegC() [2/2]	145
5.4.1.717 operator"" "" _mdegC() [1/2]	145

5.4.1.718 operator"" "" _mdegC() [2/2]	145
5.4.1.719 operator"" "" _mF() [1/2]	146
5.4.1.720 operator"" "" _MF() [1/2]	146
5.4.1.721 operator"" "" _MF() [2/2]	146
5.4.1.722 operator"" "" _mF() [2/2]	146
5.4.1.723 operator"" "" _mg() [1/2]	146
5.4.1.724 operator"" "" _mg() [2/2]	146
5.4.1.725 operator"" "" _Mg() [1/2]	146
5.4.1.726 operator"" "" _Mg() [2/2]	147
5.4.1.727 operator"" "" _MGy() [1/2]	147
5.4.1.728 operator"" "" _mGy() [1/2]	147
5.4.1.729 operator"" "" _mGy() [2/2]	147
5.4.1.730 operator"" "" _MGy() [2/2]	147
5.4.1.731 operator"" "" _mH() [1/2]	147
5.4.1.732 operator"" "" _MH() [1/2]	147
5.4.1.733 operator"" "" _MH() [2/2]	148
5.4.1.734 operator"" "" _mH() [2/2]	148
5.4.1.735 operator"" "" _Mh() [1/2]	148
5.4.1.736 operator"" "" _mh() [1/2]	148
5.4.1.737 operator"" "" _mh() [2/2]	148
5.4.1.738 operator"" "" _Mh() [2/2]	148
5.4.1.739 operator"" "" _MHz() [1/2]	148
5.4.1.740 operator"" "" _mHz() [1/2]	149
5.4.1.741 operator"" "" _mHz() [2/2]	149
5.4.1.742 operator"" "" _MHz() [2/2]	149
5.4.1.743 operator"" "" _min() [1/2]	149
5.4.1.744 operator"" "" _min() [2/2]	149
5.4.1.745 operator"" "" _MJ() [1/2]	149
5.4.1.746 operator"" "" _MJ() [2/2]	149
5.4.1.747 operator"" "" _mJ() [1/2]	150

5.4.1.748 operator"" "" _mJ() [2/2]	150
5.4.1.749 operator"" "" _MK() [1/2]	150
5.4.1.750 operator"" "" _mK() [1/2]	150
5.4.1.751 operator"" "" _mK() [2/2]	150
5.4.1.752 operator"" "" _MK() [2/2]	150
5.4.1.753 operator"" "" _Mkat() [1/2]	150
5.4.1.754 operator"" "" _mkat() [1/2]	151
5.4.1.755 operator"" "" _mkat() [2/2]	151
5.4.1.756 operator"" "" _Mkat() [2/2]	151
5.4.1.757 operator"" "" _MI() [1/2]	151
5.4.1.758 operator"" "" _ml() [1/2]	151
5.4.1.759 operator"" "" _ml() [2/2]	151
5.4.1.760 operator"" "" _MI() [2/2]	151
5.4.1.761 operator"" "" _ML() [1/2]	152
5.4.1.762 operator"" "" _mL() [1/2]	152
5.4.1.763 operator"" "" _mL() [2/2]	152
5.4.1.764 operator"" "" _ML() [2/2]	152
5.4.1.765 operator"" "" _Mlm() [1/2]	152
5.4.1.766 operator"" "" _Mlm() [2/2]	152
5.4.1.767 operator"" "" _mlm() [1/2]	152
5.4.1.768 operator"" "" _mlm() [2/2]	153
5.4.1.769 operator"" "" _mlx() [1/2]	153
5.4.1.770 operator"" "" _mlx() [2/2]	153
5.4.1.771 operator"" "" _Mlx() [1/2]	153
5.4.1.772 operator"" "" _Mlx() [2/2]	153
5.4.1.773 operator"" "" _Mm() [1/2]	153
5.4.1.774 operator"" "" _Mm() [2/2]	153
5.4.1.775 operator"" "" _mm() [1/2]	154
5.4.1.776 operator"" "" _mm() [2/2]	154
5.4.1.777 operator"" "" _mmin() [1/2]	154

5.4.1.778 operator"" "" _Mmin() [1/2]	154
5.4.1.779 operator"" "" _Mmin() [2/2]	154
5.4.1.780 operator"" "" _mmin() [2/2]	154
5.4.1.781 operator"" "" _Mmol() [1/2]	154
5.4.1.782 operator"" "" _Mmol() [2/2]	155
5.4.1.783 operator"" "" _mmol() [1/2]	155
5.4.1.784 operator"" "" _mmol() [2/2]	155
5.4.1.785 operator"" "" _MN() [1/2]	155
5.4.1.786 operator"" "" _MN() [2/2]	155
5.4.1.787 operator"" "" _mN() [1/2]	155
5.4.1.788 operator"" "" _mN() [2/2]	155
5.4.1.789 operator"" "" _Mohm() [1/2]	156
5.4.1.790 operator"" "" _mohm() [1/2]	156
5.4.1.791 operator"" "" _Mohm() [2/2]	156
5.4.1.792 operator"" "" _mohm() [2/2]	156
5.4.1.793 operator"" "" _mol() [1/2]	156
5.4.1.794 operator"" "" _mol() [2/2]	156
5.4.1.795 operator"" "" _mPa() [1/2]	156
5.4.1.796 operator"" "" _MPa() [1/2]	157
5.4.1.797 operator"" "" _MPa() [2/2]	157
5.4.1.798 operator"" "" _mPa() [2/2]	157
5.4.1.799 operator"" "" _Mrad() [1/2]	157
5.4.1.800 operator"" "" _mrad() [1/2]	157
5.4.1.801 operator"" "" _mrad() [2/2]	157
5.4.1.802 operator"" "" _Mrad() [2/2]	157
5.4.1.803 operator"" "" _Ms() [1/2]	158
5.4.1.804 operator"" "" _Ms() [2/2]	158
5.4.1.805 operator"" "" _ms() [1/2]	158
5.4.1.806 operator"" "" _ms() [2/2]	158
5.4.1.807 operator"" "" _MS() [1/2]	158

5.4.1.808 operator"" _MS() [2/2]	158
5.4.1.809 operator"" _mS() [1/2]	158
5.4.1.810 operator"" _mS() [2/2]	159
5.4.1.811 operator"" _Msr() [1/2]	159
5.4.1.812 operator"" _Msr() [2/2]	159
5.4.1.813 operator"" _msr() [1/2]	159
5.4.1.814 operator"" _msr() [2/2]	159
5.4.1.815 operator"" _mSv() [1/2]	159
5.4.1.816 operator"" _mSv() [2/2]	159
5.4.1.817 operator"" _MSv() [1/2]	160
5.4.1.818 operator"" _MSv() [2/2]	160
5.4.1.819 operator"" _MT() [1/2]	160
5.4.1.820 operator"" _MT() [2/2]	160
5.4.1.821 operator"" _mT() [1/2]	160
5.4.1.822 operator"" _mT() [2/2]	160
5.4.1.823 operator"" _Mt() [1/2]	160
5.4.1.824 operator"" _mt() [1/2]	161
5.4.1.825 operator"" _Mt() [2/2]	161
5.4.1.826 operator"" _mt() [2/2]	161
5.4.1.827 operator"" _MV() [1/2]	161
5.4.1.828 operator"" _MV() [2/2]	161
5.4.1.829 operator"" _mV() [1/2]	161
5.4.1.830 operator"" _mV() [2/2]	161
5.4.1.831 operator"" _MW() [1/2]	162
5.4.1.832 operator"" _mW() [1/2]	162
5.4.1.833 operator"" _MW() [2/2]	162
5.4.1.834 operator"" _mW() [2/2]	162
5.4.1.835 operator"" _MWb() [1/2]	162
5.4.1.836 operator"" _MWb() [2/2]	162
5.4.1.837 operator"" _mWb() [1/2]	162

5.4.1.838 operator"" "" _mWb() [2/2]	163
5.4.1.839 operator"" "" _N() [1/2]	163
5.4.1.840 operator"" "" _N() [2/2]	163
5.4.1.841 operator"" "" _nA() [1/2]	163
5.4.1.842 operator"" "" _nA() [2/2]	163
5.4.1.843 operator"" "" _nBq() [1/2]	163
5.4.1.844 operator"" "" _nBq() [2/2]	163
5.4.1.845 operator"" "" _nC() [1/2]	164
5.4.1.846 operator"" "" _nC() [2/2]	164
5.4.1.847 operator"" "" _ncd() [1/2]	164
5.4.1.848 operator"" "" _ncd() [2/2]	164
5.4.1.849 operator"" "" _nday() [1/2]	164
5.4.1.850 operator"" "" _nday() [2/2]	164
5.4.1.851 operator"" "" _ndeg() [1/2]	164
5.4.1.852 operator"" "" _ndeg() [2/2]	165
5.4.1.853 operator"" "" _ndegC() [1/2]	165
5.4.1.854 operator"" "" _ndegC() [2/2]	165
5.4.1.855 operator"" "" _nF() [1/2]	165
5.4.1.856 operator"" "" _nF() [2/2]	165
5.4.1.857 operator"" "" _ng() [1/2]	165
5.4.1.858 operator"" "" _ng() [2/2]	165
5.4.1.859 operator"" "" _nGy() [1/2]	166
5.4.1.860 operator"" "" _nGy() [2/2]	166
5.4.1.861 operator"" "" _nH() [1/2]	166
5.4.1.862 operator"" "" _nH() [2/2]	166
5.4.1.863 operator"" "" _nh() [1/2]	166
5.4.1.864 operator"" "" _nh() [2/2]	166
5.4.1.865 operator"" "" _nHz() [1/2]	166
5.4.1.866 operator"" "" _nHz() [2/2]	167
5.4.1.867 operator"" "" _nJ() [1/2]	167

5.4.1.868 operator"" "" _nJ() [2/2]	167
5.4.1.869 operator"" "" _nK() [1/2]	167
5.4.1.870 operator"" "" _nK() [2/2]	167
5.4.1.871 operator"" "" _nkat() [1/2]	167
5.4.1.872 operator"" "" _nkat() [2/2]	167
5.4.1.873 operator"" "" _nl() [1/2]	168
5.4.1.874 operator"" "" _nl() [2/2]	168
5.4.1.875 operator"" "" _nL() [1/2]	168
5.4.1.876 operator"" "" _nL() [2/2]	168
5.4.1.877 operator"" "" _nlm() [1/2]	168
5.4.1.878 operator"" "" _nlm() [2/2]	168
5.4.1.879 operator"" "" _nlx() [1/2]	168
5.4.1.880 operator"" "" _nlx() [2/2]	169
5.4.1.881 operator"" "" _nm() [1/2]	169
5.4.1.882 operator"" "" _nm() [2/2]	169
5.4.1.883 operator"" "" _nmin() [1/2]	169
5.4.1.884 operator"" "" _nmin() [2/2]	169
5.4.1.885 operator"" "" _nmol() [1/2]	169
5.4.1.886 operator"" "" _nmol() [2/2]	169
5.4.1.887 operator"" "" _nN() [1/2]	170
5.4.1.888 operator"" "" _nN() [2/2]	170
5.4.1.889 operator"" "" _nohm() [1/2]	170
5.4.1.890 operator"" "" _nohm() [2/2]	170
5.4.1.891 operator"" "" _nP() [1/2]	170
5.4.1.892 operator"" "" _nP() [2/2]	170
5.4.1.893 operator"" "" _nrad() [1/2]	170
5.4.1.894 operator"" "" _nrad() [2/2]	171
5.4.1.895 operator"" "" _ns() [1/2]	171
5.4.1.896 operator"" "" _ns() [2/2]	171
5.4.1.897 operator"" "" _nS() [1/2]	171

5.4.1.898 operator"" "" _nS() [2/2]	171
5.4.1.899 operator"" "" _nsr() [1/2]	171
5.4.1.900 operator"" "" _nsr() [2/2]	171
5.4.1.901 operator"" "" _nSv() [1/2]	172
5.4.1.902 operator"" "" _nSv() [2/2]	172
5.4.1.903 operator"" "" _nT() [1/2]	172
5.4.1.904 operator"" "" _nT() [2/2]	172
5.4.1.905 operator"" "" _nt() [1/2]	172
5.4.1.906 operator"" "" _nt() [2/2]	172
5.4.1.907 operator"" "" _nV() [1/2]	172
5.4.1.908 operator"" "" _nV() [2/2]	173
5.4.1.909 operator"" "" _nW() [1/2]	173
5.4.1.910 operator"" "" _nW() [2/2]	173
5.4.1.911 operator"" "" _nWb() [1/2]	173
5.4.1.912 operator"" "" _nWb() [2/2]	173
5.4.1.913 operator"" "" _ohm() [1/2]	173
5.4.1.914 operator"" "" _ohm() [2/2]	173
5.4.1.915 operator"" "" _pA() [1/2]	174
5.4.1.916 operator"" "" _pA() [2/2]	174
5.4.1.917 operator"" "" _PA() [1/2]	174
5.4.1.918 operator"" "" _PA() [2/2]	174
5.4.1.919 operator"" "" _Pa() [1/2]	174
5.4.1.920 operator"" "" _Pa() [2/2]	174
5.4.1.921 operator"" "" _PBq() [1/2]	174
5.4.1.922 operator"" "" _PBq() [2/2]	175
5.4.1.923 operator"" "" _pBq() [1/2]	175
5.4.1.924 operator"" "" _pBq() [2/2]	175
5.4.1.925 operator"" "" _PC() [1/2]	175
5.4.1.926 operator"" "" _pC() [1/2]	175
5.4.1.927 operator"" "" _pC() [2/2]	175

5.4.1.928 operator"" _PC() [2/2]	175
5.4.1.929 operator"" _Pcd() [1/2]	176
5.4.1.930 operator"" _pcd() [1/2]	176
5.4.1.931 operator"" _pcd() [2/2]	176
5.4.1.932 operator"" _Pcd() [2/2]	176
5.4.1.933 operator"" _pday() [1/2]	176
5.4.1.934 operator"" _Pday() [1/2]	176
5.4.1.935 operator"" _Pday() [2/2]	176
5.4.1.936 operator"" _pday() [2/2]	177
5.4.1.937 operator"" _Pdeg() [1/2]	177
5.4.1.938 operator"" _Pdeg() [2/2]	177
5.4.1.939 operator"" _pdeg() [1/2]	177
5.4.1.940 operator"" _pdeg() [2/2]	177
5.4.1.941 operator"" _pdegC() [1/2]	177
5.4.1.942 operator"" _pdegC() [2/2]	177
5.4.1.943 operator"" _PdegC() [1/2]	178
5.4.1.944 operator"" _PdegC() [2/2]	178
5.4.1.945 operator"" _pF() [1/2]	178
5.4.1.946 operator"" _PF() [1/2]	178
5.4.1.947 operator"" _PF() [2/2]	178
5.4.1.948 operator"" _pF() [2/2]	178
5.4.1.949 operator"" _pg() [1/2]	178
5.4.1.950 operator"" _pg() [2/2]	179
5.4.1.951 operator"" _Pg() [1/2]	179
5.4.1.952 operator"" _Pg() [2/2]	179
5.4.1.953 operator"" _PGy() [1/2]	179
5.4.1.954 operator"" _PGy() [2/2]	179
5.4.1.955 operator"" _pGy() [1/2]	179
5.4.1.956 operator"" _pGy() [2/2]	179
5.4.1.957 operator"" _PH() [1/2]	180

5.4.1.958 operator"" "" _PH() [2/2]	180
5.4.1.959 operator"" "" _pH() [1/2]	180
5.4.1.960 operator"" "" _pH() [2/2]	180
5.4.1.961 operator"" "" _Ph() [1/2]	180
5.4.1.962 operator"" "" _Ph() [2/2]	180
5.4.1.963 operator"" "" _ph() [1/2]	180
5.4.1.964 operator"" "" _ph() [2/2]	181
5.4.1.965 operator"" "" _pHz() [1/2]	181
5.4.1.966 operator"" "" _PHz() [1/2]	181
5.4.1.967 operator"" "" _PHz() [2/2]	181
5.4.1.968 operator"" "" _pHz() [2/2]	181
5.4.1.969 operator"" "" _pJ() [1/2]	181
5.4.1.970 operator"" "" _PJ() [1/2]	181
5.4.1.971 operator"" "" _pJ() [2/2]	182
5.4.1.972 operator"" "" _PJ() [2/2]	182
5.4.1.973 operator"" "" _PK() [1/2]	182
5.4.1.974 operator"" "" _PK() [2/2]	182
5.4.1.975 operator"" "" _pK() [1/2]	182
5.4.1.976 operator"" "" _pK() [2/2]	182
5.4.1.977 operator"" "" _Pkat() [1/2]	182
5.4.1.978 operator"" "" _Pkat() [2/2]	183
5.4.1.979 operator"" "" _pkat() [1/2]	183
5.4.1.980 operator"" "" _pkat() [2/2]	183
5.4.1.981 operator"" "" _Pl() [1/2]	183
5.4.1.982 operator"" "" _Pl() [2/2]	183
5.4.1.983 operator"" "" _pl() [1/2]	183
5.4.1.984 operator"" "" _pl() [2/2]	183
5.4.1.985 operator"" "" _PL() [1/2]	184
5.4.1.986 operator"" "" _PL() [2/2]	184
5.4.1.987 operator"" "" _pL() [1/2]	184

5.4.1.988 operator"" "" _pL() [2/2]	184
5.4.1.989 operator"" "" _plm() [1/2]	184
5.4.1.990 operator"" "" _Plm() [1/2]	184
5.4.1.991 operator"" "" _Plm() [2/2]	184
5.4.1.992 operator"" "" _plm() [2/2]	185
5.4.1.993 operator"" "" _plx() [1/2]	185
5.4.1.994 operator"" "" _Plx() [1/2]	185
5.4.1.995 operator"" "" _Plx() [2/2]	185
5.4.1.996 operator"" "" _plx() [2/2]	185
5.4.1.997 operator"" "" _pm() [1/2]	185
5.4.1.998 operator"" "" _Pm() [1/2]	185
5.4.1.999 operator"" "" _pm() [2/2]	186
5.4.1.100operator"" "" _Pm() [2/2]	186
5.4.1.1001operator"" "" _Pmin() [1/2]	186
5.4.1.1002operator"" "" _pmin() [1/2]	186
5.4.1.1003operator"" "" _pmin() [2/2]	186
5.4.1.1004operator"" "" _Pmin() [2/2]	186
5.4.1.1005operator"" "" _Pmol() [1/2]	186
5.4.1.1006operator"" "" _Pmol() [2/2]	187
5.4.1.1007operator"" "" _pmol() [1/2]	187
5.4.1.1008operator"" "" _pmol() [2/2]	187
5.4.1.1009operator"" "" _pN() [1/2]	187
5.4.1.1010operator"" "" _pN() [2/2]	187
5.4.1.1011operator"" "" _PN() [1/2]	187
5.4.1.1012operator"" "" _PN() [2/2]	187
5.4.1.1013operator"" "" _pohm() [1/2]	188
5.4.1.1014operator"" "" _pohm() [2/2]	188
5.4.1.1015operator"" "" _Pohm() [1/2]	188
5.4.1.1016operator"" "" _Pohm() [2/2]	188
5.4.1.1017operator"" "" _PPa() [1/2]	188

5.4.1.1018operator"" "" _pPa() [1/2]	188
5.4.1.1019operator"" "" _PPa() [2/2]	188
5.4.1.1020operator"" "" _pPa() [2/2]	189
5.4.1.1021operator"" "" _prad() [1/2]	189
5.4.1.1022operator"" "" _Prad() [1/2]	189
5.4.1.1023operator"" "" _Prad() [2/2]	189
5.4.1.1024operator"" "" _prad() [2/2]	189
5.4.1.1025operator"" "" _Ps() [1/2]	189
5.4.1.1026operator"" "" _Ps() [2/2]	189
5.4.1.1027operator"" "" _ps() [1/2]	190
5.4.1.1028operator"" "" _ps() [2/2]	190
5.4.1.1029operator"" "" _PS() [1/2]	190
5.4.1.1030operator"" "" _PS() [2/2]	190
5.4.1.1031operator"" "" _pS() [1/2]	190
5.4.1.1032operator"" "" _pS() [2/2]	190
5.4.1.1033operator"" "" _psr() [1/2]	190
5.4.1.1034operator"" "" _Psr() [1/2]	191
5.4.1.1035operator"" "" _Psr() [2/2]	191
5.4.1.1036operator"" "" _psr() [2/2]	191
5.4.1.1037operator"" "" _PSv() [1/2]	191
5.4.1.1038operator"" "" _PSv() [2/2]	191
5.4.1.1039operator"" "" _pSv() [1/2]	191
5.4.1.1040operator"" "" _pSv() [2/2]	191
5.4.1.1041operator"" "" _pT() [1/2]	192
5.4.1.1042operator"" "" _PT() [1/2]	192
5.4.1.1043operator"" "" _PT() [2/2]	192
5.4.1.1044operator"" "" _pT() [2/2]	192
5.4.1.1045operator"" "" _pt() [1/2]	192
5.4.1.1046operator"" "" _Pt() [1/2]	192
5.4.1.1047operator"" "" _pt() [2/2]	192

5.4.1.1048operator"" "" _Pt() [2/2]	193
5.4.1.1049operator"" "" _pV() [1/2]	193
5.4.1.1050operator"" "" _pV() [2/2]	193
5.4.1.1051operator"" "" _PV() [1/2]	193
5.4.1.1052operator"" "" _PV() [2/2]	193
5.4.1.1053operator"" "" _PW() [1/2]	193
5.4.1.1054operator"" "" _PW() [2/2]	193
5.4.1.1055operator"" "" _pW() [1/2]	194
5.4.1.1056operator"" "" _pW() [2/2]	194
5.4.1.1057operator"" "" _pWb() [1/2]	194
5.4.1.1058operator"" "" _PWB() [1/2]	194
5.4.1.1059operator"" "" _PWB() [2/2]	194
5.4.1.1060operator"" "" _pWb() [2/2]	194
5.4.1.1061operator"" "" _rad() [1/2]	194
5.4.1.1062operator"" "" _rad() [2/2]	195
5.4.1.1063operator"" "" _s() [1/2]	195
5.4.1.1064operator"" "" _s() [2/2]	195
5.4.1.1065operator"" "" _S() [1/2]	195
5.4.1.1066operator"" "" _S() [2/2]	195
5.4.1.1067operator"" "" _sr() [1/2]	195
5.4.1.1068operator"" "" _sr() [2/2]	195
5.4.1.1069operator"" "" _Sv() [1/2]	196
5.4.1.1070operator"" "" _Sv() [2/2]	196
5.4.1.1071operator"" "" _T() [1/2]	196
5.4.1.1072operator"" "" _T() [2/2]	196
5.4.1.1073operator"" "" _t() [1/2]	196
5.4.1.1074operator"" "" _t() [2/2]	196
5.4.1.1075operator"" "" _TA() [1/2]	196
5.4.1.1076operator"" "" _TA() [2/2]	197
5.4.1.1077operator"" "" _TBq() [1/2]	197

5.4.1.107operator"" _TBq() [2/2]	197
5.4.1.1079operator"" _TC() [1/2]	197
5.4.1.1080operator"" _TC() [2/2]	197
5.4.1.1081operator"" _Tcd() [1/2]	197
5.4.1.1082operator"" _Tcd() [2/2]	197
5.4.1.1083operator"" _Tday() [1/2]	198
5.4.1.1084operator"" _Tday() [2/2]	198
5.4.1.1085operator"" _Tdeg() [1/2]	198
5.4.1.1086operator"" _Tdeg() [2/2]	198
5.4.1.1087operator"" _TdegC() [1/2]	198
5.4.1.1088operator"" _TdegC() [2/2]	198
5.4.1.1089operator"" _TF() [1/2]	198
5.4.1.1090operator"" _TF() [2/2]	199
5.4.1.1091operator"" _Tg() [1/2]	199
5.4.1.1092operator"" _Tg() [2/2]	199
5.4.1.1093operator"" _TGy() [1/2]	199
5.4.1.1094operator"" _TGy() [2/2]	199
5.4.1.1095operator"" _TH() [1/2]	199
5.4.1.1096operator"" _TH() [2/2]	199
5.4.1.1097operator"" _Th() [1/2]	200
5.4.1.1098operator"" _Th() [2/2]	200
5.4.1.1099operator"" _THz() [1/2]	200
5.4.1.1100operator"" _THz() [2/2]	200
5.4.1.1101operator"" _TJ() [1/2]	200
5.4.1.1102operator"" _TJ() [2/2]	200
5.4.1.1103operator"" _TK() [1/2]	200
5.4.1.1104operator"" _TK() [2/2]	201
5.4.1.1105operator"" _Tkat() [1/2]	201
5.4.1.1106operator"" _Tkat() [2/2]	201
5.4.1.1107operator"" _Tl() [1/2]	201

5.4.1.110operator"" _Tl() [2/2]	201
5.4.1.110operator"" _Tl() [1/2]	201
5.4.1.110operator"" _Tl() [2/2]	201
5.4.1.111operator"" _Tlm() [1/2]	202
5.4.1.112operator"" _Tlm() [2/2]	202
5.4.1.113operator"" _Tlx() [1/2]	202
5.4.1.114operator"" _Tlx() [2/2]	202
5.4.1.115operator"" _Tm() [1/2]	202
5.4.1.116operator"" _Tm() [2/2]	202
5.4.1.117operator"" _Tmin() [1/2]	202
5.4.1.118operator"" _Tmin() [2/2]	203
5.4.1.119operator"" _Tmol() [1/2]	203
5.4.1.1120operator"" _Tmol() [2/2]	203
5.4.1.1121operator"" _TN() [1/2]	203
5.4.1.1122operator"" _TN() [2/2]	203
5.4.1.1123operator"" _Tohm() [1/2]	203
5.4.1.1124operator"" _Tohm() [2/2]	203
5.4.1.1125operator"" _TPa() [1/2]	204
5.4.1.1126operator"" _TPa() [2/2]	204
5.4.1.1127operator"" _Trad() [1/2]	204
5.4.1.1128operator"" _Trad() [2/2]	204
5.4.1.1129operator"" _Ts() [1/2]	204
5.4.1.1130operator"" _Ts() [2/2]	204
5.4.1.1131operator"" _TS() [1/2]	204
5.4.1.1132operator"" _TS() [2/2]	205
5.4.1.1133operator"" _Tsr() [1/2]	205
5.4.1.1134operator"" _Tsr() [2/2]	205
5.4.1.1135operator"" _TSv() [1/2]	205
5.4.1.1136operator"" _TSv() [2/2]	205
5.4.1.1137operator"" _TT() [1/2]	205

5.4.1.113operator"" _TT() [2/2]	205
5.4.1.113operator"" _Tt() [1/2]	206
5.4.1.114operator"" _Tt() [2/2]	206
5.4.1.114operator"" _TV() [1/2]	206
5.4.1.114operator"" _TV() [2/2]	206
5.4.1.114operator"" _TW() [1/2]	206
5.4.1.114operator"" _TW() [2/2]	206
5.4.1.114operator"" _TWb() [1/2]	206
5.4.1.114operator"" _TWb() [2/2]	207
5.4.1.114operator"" _uA() [1/2]	207
5.4.1.114operator"" _uA() [2/2]	207
5.4.1.114operator"" _uBq() [1/2]	207
5.4.1.115operator"" _uBq() [2/2]	207
5.4.1.115operator"" _uC() [1/2]	207
5.4.1.115operator"" _uC() [2/2]	207
5.4.1.115operator"" _ucd() [1/2]	208
5.4.1.115operator"" _ucd() [2/2]	208
5.4.1.115operator"" _uday() [1/2]	208
5.4.1.115operator"" _uday() [2/2]	208
5.4.1.115operator"" _udeg() [1/2]	208
5.4.1.115operator"" _udeg() [2/2]	208
5.4.1.115operator"" _udegC() [1/2]	208
5.4.1.115operator"" _udegC() [2/2]	209
5.4.1.116operator"" _uF() [1/2]	209
5.4.1.116operator"" _uF() [2/2]	209
5.4.1.116operator"" _ug() [1/2]	209
5.4.1.116operator"" _ug() [2/2]	209
5.4.1.116operator"" _uGy() [1/2]	209
5.4.1.116operator"" _uGy() [2/2]	209
5.4.1.116operator"" _uH() [1/2]	210

5.4.1.1168operator"" "" _uH() [2/2]	210
5.4.1.1169operator"" "" _uh() [1/2]	210
5.4.1.1170operator"" "" _uh() [2/2]	210
5.4.1.1171operator"" "" _uHz() [1/2]	210
5.4.1.1172operator"" "" _uHz() [2/2]	210
5.4.1.1173operator"" "" _uJ() [1/2]	210
5.4.1.1174operator"" "" _uJ() [2/2]	211
5.4.1.1175operator"" "" _uK() [1/2]	211
5.4.1.1176operator"" "" _uK() [2/2]	211
5.4.1.1177operator"" "" _ukat() [1/2]	211
5.4.1.1178operator"" "" _ukat() [2/2]	211
5.4.1.1179operator"" "" _ul() [1/2]	211
5.4.1.1180operator"" "" _ul() [2/2]	211
5.4.1.1181operator"" "" _uL() [1/2]	212
5.4.1.1182operator"" "" _uL() [2/2]	212
5.4.1.1183operator"" "" _ulm() [1/2]	212
5.4.1.1184operator"" "" _ulm() [2/2]	212
5.4.1.1185operator"" "" _ulx() [1/2]	212
5.4.1.1186operator"" "" _ulx() [2/2]	212
5.4.1.1187operator"" "" _um() [1/2]	212
5.4.1.1188operator"" "" _um() [2/2]	213
5.4.1.1189operator"" "" _umin() [1/2]	213
5.4.1.1190operator"" "" _umin() [2/2]	213
5.4.1.1191operator"" "" _umol() [1/2]	213
5.4.1.1192operator"" "" _umol() [2/2]	213
5.4.1.1193operator"" "" _uN() [1/2]	213
5.4.1.1194operator"" "" _uN() [2/2]	213
5.4.1.1195operator"" "" _uohm() [1/2]	214
5.4.1.1196operator"" "" _uohm() [2/2]	214
5.4.1.1197operator"" "" _uPa() [1/2]	214

5.4.1.119operator"" "" _uPa() [2/2]	214
5.4.1.1199operator"" "" _urad() [1/2]	214
5.4.1.1200operator"" "" _urad() [2/2]	214
5.4.1.1201operator"" "" _us() [1/2]	214
5.4.1.1202operator"" "" _us() [2/2]	215
5.4.1.1203operator"" "" _uS() [1/2]	215
5.4.1.1204operator"" "" _uS() [2/2]	215
5.4.1.1205operator"" "" _usr() [1/2]	215
5.4.1.1206operator"" "" _usr() [2/2]	215
5.4.1.1207operator"" "" _uSv() [1/2]	215
5.4.1.1208operator"" "" _uSv() [2/2]	215
5.4.1.1209operator"" "" _uT() [1/2]	216
5.4.1.1210operator"" "" _uT() [2/2]	216
5.4.1.1211operator"" "" _ut() [1/2]	216
5.4.1.1212operator"" "" _ut() [2/2]	216
5.4.1.1213operator"" "" _uV() [1/2]	216
5.4.1.1214operator"" "" _uV() [2/2]	216
5.4.1.1215operator"" "" _uW() [1/2]	216
5.4.1.1216operator"" "" _uW() [2/2]	217
5.4.1.1217operator"" "" _uWb() [1/2]	217
5.4.1.1218operator"" "" _uWb() [2/2]	217
5.4.1.1219operator"" "" _V() [1/2]	217
5.4.1.1220operator"" "" _V() [2/2]	217
5.4.1.1221operator"" "" _W() [1/2]	217
5.4.1.1222operator"" "" _W() [2/2]	217
5.4.1.1223operator"" "" _Wb() [1/2]	218
5.4.1.1224operator"" "" _Wb() [2/2]	218
5.4.1.1225operator"" "" _YA() [1/2]	218
5.4.1.1226operator"" "" _yA() [1/2]	218
5.4.1.1227operator"" "" _YA() [2/2]	218

5.4.1.1228operator"" "" _yA() [2/2]	218
5.4.1.1229operator"" "" _YBq() [1/2]	218
5.4.1.1230operator"" "" _yBq() [1/2]	219
5.4.1.1231operator"" "" _YBq() [2/2]	219
5.4.1.1232operator"" "" _yBq() [2/2]	219
5.4.1.1233operator"" "" _YC() [1/2]	219
5.4.1.1234operator"" "" _YC() [2/2]	219
5.4.1.1235operator"" "" _yC() [1/2]	219
5.4.1.1236operator"" "" _yC() [2/2]	219
5.4.1.1237operator"" "" _ycd() [1/2]	220
5.4.1.1238operator"" "" _ycd() [2/2]	220
5.4.1.1239operator"" "" _Ycd() [1/2]	220
5.4.1.1240operator"" "" _Ycd() [2/2]	220
5.4.1.1241operator"" "" _Yday() [1/2]	220
5.4.1.1242operator"" "" _yday() [1/2]	220
5.4.1.1243operator"" "" _yday() [2/2]	220
5.4.1.1244operator"" "" _Yday() [2/2]	221
5.4.1.1245operator"" "" _Ydeg() [1/2]	221
5.4.1.1246operator"" "" _ydeg() [1/2]	221
5.4.1.1247operator"" "" _Ydeg() [2/2]	221
5.4.1.1248operator"" "" _ydeg() [2/2]	221
5.4.1.1249operator"" "" _ydegC() [1/2]	221
5.4.1.1250operator"" "" _YdegC() [1/2]	221
5.4.1.1251operator"" "" _YdegC() [2/2]	222
5.4.1.1252operator"" "" _ydegC() [2/2]	222
5.4.1.1253operator"" "" _YF() [1/2]	222
5.4.1.1254operator"" "" _YF() [2/2]	222
5.4.1.1255operator"" "" _yF() [1/2]	222
5.4.1.1256operator"" "" _yF() [2/2]	222
5.4.1.1257operator"" "" _yg() [1/2]	222

5.4.1.1258operator"" "" _Yg() [1/2]	223
5.4.1.1259operator"" "" _Yg() [2/2]	223
5.4.1.1260operator"" "" _yg() [2/2]	223
5.4.1.1261operator"" "" _yGy() [1/2]	223
5.4.1.1262operator"" "" _yGy() [2/2]	223
5.4.1.1263operator"" "" _YGy() [1/2]	223
5.4.1.1264operator"" "" _YGy() [2/2]	223
5.4.1.1265operator"" "" _yH() [1/2]	224
5.4.1.1266operator"" "" _YH() [1/2]	224
5.4.1.1267operator"" "" _YH() [2/2]	224
5.4.1.1268operator"" "" _yH() [2/2]	224
5.4.1.1269operator"" "" _Yh() [1/2]	224
5.4.1.1270operator"" "" _yh() [1/2]	224
5.4.1.1271operator"" "" _yh() [2/2]	224
5.4.1.1272operator"" "" _Yh() [2/2]	225
5.4.1.1273operator"" "" _YHz() [1/2]	225
5.4.1.1274operator"" "" _yHz() [1/2]	225
5.4.1.1275operator"" "" _yHz() [2/2]	225
5.4.1.1276operator"" "" _YHz() [2/2]	225
5.4.1.1277operator"" "" _yJ() [1/2]	225
5.4.1.1278operator"" "" _yJ() [2/2]	225
5.4.1.1279operator"" "" _YJ() [1/2]	226
5.4.1.1280operator"" "" _YJ() [2/2]	226
5.4.1.1281operator"" "" _YK() [1/2]	226
5.4.1.1282operator"" "" _yK() [1/2]	226
5.4.1.1283operator"" "" _yK() [2/2]	226
5.4.1.1284operator"" "" _YK() [2/2]	226
5.4.1.1285operator"" "" _ykat() [1/2]	226
5.4.1.1286operator"" "" _Ykat() [1/2]	227
5.4.1.1287operator"" "" _ykat() [2/2]	227

5.4.1.1288operator"" "" _Ykat() [2/2]	227
5.4.1.1289operator"" "" _Yl() [1/2]	227
5.4.1.1290operator"" "" _yl() [1/2]	227
5.4.1.1291operator"" "" _yl() [2/2]	227
5.4.1.1292operator"" "" _Yl() [2/2]	227
5.4.1.1293operator"" "" _yL() [1/2]	228
5.4.1.1294operator"" "" _YL() [1/2]	228
5.4.1.1295operator"" "" _YL() [2/2]	228
5.4.1.1296operator"" "" _yL() [2/2]	228
5.4.1.1297operator"" "" _Ylm() [1/2]	228
5.4.1.1298operator"" "" _Ylm() [2/2]	228
5.4.1.1299operator"" "" _ylm() [1/2]	228
5.4.1.1300operator"" "" _ylm() [2/2]	229
5.4.1.1301operator"" "" _Ylx() [1/2]	229
5.4.1.1302operator"" "" _Ylx() [2/2]	229
5.4.1.1303operator"" "" _ylx() [1/2]	229
5.4.1.1304operator"" "" _ylx() [2/2]	229
5.4.1.1305operator"" "" _Ym() [1/2]	229
5.4.1.1306operator"" "" _ym() [1/2]	229
5.4.1.1307operator"" "" _ym() [2/2]	230
5.4.1.1308operator"" "" _Ym() [2/2]	230
5.4.1.1309operator"" "" _ymin() [1/2]	230
5.4.1.1310operator"" "" _Ymin() [1/2]	230
5.4.1.1311operator"" "" _ymin() [2/2]	230
5.4.1.1312operator"" "" _Ymin() [2/2]	230
5.4.1.1313operator"" "" _Ymol() [1/2]	230
5.4.1.1314operator"" "" _ymol() [1/2]	231
5.4.1.1315operator"" "" _Ymol() [2/2]	231
5.4.1.1316operator"" "" _ymol() [2/2]	231
5.4.1.1317operator"" "" _YN() [1/2]	231

5.4.1.1318operator"" "" _YN() [2/2]	231
5.4.1.1319operator"" "" _yN() [1/2]	231
5.4.1.1320operator"" "" _yN() [2/2]	231
5.4.1.1321operator"" "" _yohm() [1/2]	232
5.4.1.1322operator"" "" _Yohm() [1/2]	232
5.4.1.1323operator"" "" _Yohm() [2/2]	232
5.4.1.1324operator"" "" _yohm() [2/2]	232
5.4.1.1325operator"" "" _yPa() [1/2]	232
5.4.1.1326operator"" "" _YPa() [1/2]	232
5.4.1.1327operator"" "" _YPa() [2/2]	232
5.4.1.1328operator"" "" _yPa() [2/2]	233
5.4.1.1329operator"" "" _Yrad() [1/2]	233
5.4.1.1330operator"" "" _yrad() [1/2]	233
5.4.1.1331operator"" "" _Yrad() [2/2]	233
5.4.1.1332operator"" "" _yrad() [2/2]	233
5.4.1.1333operator"" "" _ys() [1/2]	233
5.4.1.1334operator"" "" _ys() [2/2]	233
5.4.1.1335operator"" "" _Ys() [1/2]	234
5.4.1.1336operator"" "" _Ys() [2/2]	234
5.4.1.1337operator"" "" _YS() [1/2]	234
5.4.1.1338operator"" "" _yS() [1/2]	234
5.4.1.1339operator"" "" _YS() [2/2]	234
5.4.1.1340operator"" "" _yS() [2/2]	234
5.4.1.1341operator"" "" _ysr() [1/2]	234
5.4.1.1342operator"" "" _ysr() [2/2]	235
5.4.1.1343operator"" "" _Ysr() [1/2]	235
5.4.1.1344operator"" "" _Ysr() [2/2]	235
5.4.1.1345operator"" "" _ySv() [1/2]	235
5.4.1.1346operator"" "" _YSv() [1/2]	235
5.4.1.1347operator"" "" _YSv() [2/2]	235

5.4.1.1348operator"" "" _ySv() [2/2]	235
5.4.1.1349operator"" "" _yT() [1/2]	236
5.4.1.1350operator"" "" _yT() [2/2]	236
5.4.1.1351operator"" "" _YT() [1/2]	236
5.4.1.1352operator"" "" _YT() [2/2]	236
5.4.1.1353operator"" "" _yt() [1/2]	236
5.4.1.1354operator"" "" _Yt() [1/2]	236
5.4.1.1355operator"" "" _Yt() [2/2]	236
5.4.1.1356operator"" "" _yt() [2/2]	237
5.4.1.1357operator"" "" _YV() [1/2]	237
5.4.1.1358operator"" "" _YV() [2/2]	237
5.4.1.1359operator"" "" _yV() [1/2]	237
5.4.1.1360operator"" "" _yV() [2/2]	237
5.4.1.1361operator"" "" _yW() [1/2]	237
5.4.1.1362operator"" "" _yW() [2/2]	237
5.4.1.1363operator"" "" _YW() [1/2]	238
5.4.1.1364operator"" "" _YW() [2/2]	238
5.4.1.1365operator"" "" _Ywb() [1/2]	238
5.4.1.1366operator"" "" _Ywb() [2/2]	238
5.4.1.1367operator"" "" _ywb() [1/2]	238
5.4.1.1368operator"" "" _ywb() [2/2]	238
5.4.1.1369operator"" "" _ZA() [1/2]	238
5.4.1.1370operator"" "" _zA() [1/2]	239
5.4.1.1371operator"" "" _ZA() [2/2]	239
5.4.1.1372operator"" "" _zA() [2/2]	239
5.4.1.1373operator"" "" _zBq() [1/2]	239
5.4.1.1374operator"" "" _ZBq() [1/2]	239
5.4.1.1375operator"" "" _zBq() [2/2]	239
5.4.1.1376operator"" "" _ZBq() [2/2]	239
5.4.1.1377operator"" "" _ZC() [1/2]	240

5.4.1.1378operator"::::: _zC() [1/2]	240
5.4.1.1379operator"::::: _ZC() [2/2]	240
5.4.1.1380operator"::::: _zC() [2/2]	240
5.4.1.1381operator"::::: _zcd() [1/2]	240
5.4.1.1382operator"::::: _Zcd() [1/2]	240
5.4.1.1383operator"::::: _zcd() [2/2]	240
5.4.1.1384operator"::::: _Zcd() [2/2]	241
5.4.1.1385operator"::::: _Zday() [1/2]	241
5.4.1.1386operator"::::: _zday() [1/2]	241
5.4.1.1387operator"::::: _zday() [2/2]	241
5.4.1.1388operator"::::: _Zday() [2/2]	241
5.4.1.1389operator"::::: _zdeg() [1/2]	241
5.4.1.1390operator"::::: _Zdeg() [1/2]	241
5.4.1.1391operator"::::: _zdeg() [2/2]	242
5.4.1.1392operator"::::: _Zdeg() [2/2]	242
5.4.1.1393operator"::::: _ZdegC() [1/2]	242
5.4.1.1394operator"::::: _zdegC() [1/2]	242
5.4.1.1395operator"::::: _ZdegC() [2/2]	242
5.4.1.1396operator"::::: _zdegC() [2/2]	242
5.4.1.1397operator"::::: _zF() [1/2]	242
5.4.1.1398operator"::::: _ZF() [1/2]	243
5.4.1.1399operator"::::: _zF() [2/2]	243
5.4.1.1400operator"::::: _ZF() [2/2]	243
5.4.1.1401operator"::::: _zg() [1/2]	243
5.4.1.1402operator"::::: _Zg() [1/2]	243
5.4.1.1403operator"::::: _Zg() [2/2]	243
5.4.1.1404operator"::::: _zg() [2/2]	243
5.4.1.1405operator"::::: _ZGy() [1/2]	244
5.4.1.1406operator"::::: _ZGy() [2/2]	244
5.4.1.1407operator"::::: _zGy() [1/2]	244

5.4.1.1408operator"" "" _zGy() [2/2]	244
5.4.1.1409operator"" "" _zH() [1/2]	244
5.4.1.1410operator"" "" _ZH() [1/2]	244
5.4.1.1411operator"" "" _ZH() [2/2]	244
5.4.1.1412operator"" "" _zH() [2/2]	245
5.4.1.1413operator"" "" _zh() [1/2]	245
5.4.1.1414operator"" "" _Zh() [1/2]	245
5.4.1.1415operator"" "" _zh() [2/2]	245
5.4.1.1416operator"" "" _Zh() [2/2]	245
5.4.1.1417operator"" "" _zHz() [1/2]	245
5.4.1.1418operator"" "" _zHz() [2/2]	245
5.4.1.1419operator"" "" _ZHz() [1/2]	246
5.4.1.1420operator"" "" _ZHz() [2/2]	246
5.4.1.1421operator"" "" _zJ() [1/2]	246
5.4.1.1422operator"" "" _ZJ() [1/2]	246
5.4.1.1423operator"" "" _ZJ() [2/2]	246
5.4.1.1424operator"" "" _zJ() [2/2]	246
5.4.1.1425operator"" "" _zK() [1/2]	246
5.4.1.1426operator"" "" _ZK() [1/2]	247
5.4.1.1427operator"" "" _zK() [2/2]	247
5.4.1.1428operator"" "" _ZK() [2/2]	247
5.4.1.1429operator"" "" _zkat() [1/2]	247
5.4.1.1430operator"" "" _zkat() [2/2]	247
5.4.1.1431operator"" "" _Zkat() [1/2]	247
5.4.1.1432operator"" "" _Zkat() [2/2]	247
5.4.1.1433operator"" "" _zl() [1/2]	248
5.4.1.1434operator"" "" _zl() [1/2]	248
5.4.1.1435operator"" "" _zl() [2/2]	248
5.4.1.1436operator"" "" _zl() [2/2]	248
5.4.1.1437operator"" "" _zL() [1/2]	248

5.4.1.1438operator"" _ZL() [1/2]	248
5.4.1.1439operator"" _zL() [2/2]	248
5.4.1.1440operator"" _ZL() [2/2]	249
5.4.1.1441operator"" _ZIm() [1/2]	249
5.4.1.1442operator"" _ZIm() [2/2]	249
5.4.1.1443operator"" _zIm() [1/2]	249
5.4.1.1444operator"" _zIm() [2/2]	249
5.4.1.1445operator"" _ZIx() [1/2]	249
5.4.1.1446operator"" _zIx() [1/2]	249
5.4.1.1447operator"" _ZIx() [2/2]	250
5.4.1.1448operator"" _zIx() [2/2]	250
5.4.1.1449operator"" _Zm() [1/2]	250
5.4.1.1450operator"" _zm() [1/2]	250
5.4.1.1451operator"" _zm() [2/2]	250
5.4.1.1452operator"" _Zm() [2/2]	250
5.4.1.1453operator"" _Zmin() [1/2]	250
5.4.1.1454operator"" _Zmin() [2/2]	251
5.4.1.1455operator"" _zmin() [1/2]	251
5.4.1.1456operator"" _zmin() [2/2]	251
5.4.1.1457operator"" _zmol() [1/2]	251
5.4.1.1458operator"" _Zmol() [1/2]	251
5.4.1.1459operator"" _zmol() [2/2]	251
5.4.1.1460operator"" _Zmol() [2/2]	251
5.4.1.1461operator"" _zN() [1/2]	252
5.4.1.1462operator"" _zN() [2/2]	252
5.4.1.1463operator"" _ZN() [1/2]	252
5.4.1.1464operator"" _ZN() [2/2]	252
5.4.1.1465operator"" _zohm() [1/2]	252
5.4.1.1466operator"" _Zohm() [1/2]	252
5.4.1.1467operator"" _zohm() [2/2]	252

5.4.1.1468operator"" _Zohm() [2/2]	253
5.4.1.1469operator"" _ZPa() [1/2]	253
5.4.1.1470operator"" _zPa() [1/2]	253
5.4.1.1471operator"" _ZPa() [2/2]	253
5.4.1.1472operator"" _zPa() [2/2]	253
5.4.1.1473operator"" _zrad() [1/2]	253
5.4.1.1474operator"" _Zrad() [1/2]	253
5.4.1.1475operator"" _Zrad() [2/2]	254
5.4.1.1476operator"" _zrad() [2/2]	254
5.4.1.1477operator"" _zs() [1/2]	254
5.4.1.1478operator"" _Zs() [1/2]	254
5.4.1.1479operator"" _Zs() [2/2]	254
5.4.1.1480operator"" _zs() [2/2]	254
5.4.1.1481operator"" _ZS() [1/2]	254
5.4.1.1482operator"" _ZS() [2/2]	255
5.4.1.1483operator"" _zS() [1/2]	255
5.4.1.1484operator"" _zS() [2/2]	255
5.4.1.1485operator"" _zsr() [1/2]	255
5.4.1.1486operator"" _zsr() [2/2]	255
5.4.1.1487operator"" _Zsr() [1/2]	255
5.4.1.1488operator"" _Zsr() [2/2]	255
5.4.1.1489operator"" _ZSv() [1/2]	256
5.4.1.1490operator"" _zSv() [1/2]	256
5.4.1.1491operator"" _ZSv() [2/2]	256
5.4.1.1492operator"" _zSv() [2/2]	256
5.4.1.1493operator"" _zT() [1/2]	256
5.4.1.1494operator"" _ZT() [1/2]	256
5.4.1.1495operator"" _zT() [2/2]	256
5.4.1.1496operator"" _ZT() [2/2]	257
5.4.1.1497operator"" _Zt() [1/2]	257

5.4.1.149operator"" "" _Zt() [2/2]	257
5.4.1.1499operator"" "" _zt() [1/2]	257
5.4.1.150operator"" "" _zt() [2/2]	257
5.4.1.1501operator"" "" _ZV() [1/2]	257
5.4.1.1502operator"" "" _zV() [1/2]	257
5.4.1.1503operator"" "" _ZV() [2/2]	258
5.4.1.1504operator"" "" _zV() [2/2]	258
5.4.1.1505operator"" "" _zW() [1/2]	258
5.4.1.1506operator"" "" _ZW() [1/2]	258
5.4.1.1507operator"" "" _zW() [2/2]	258
5.4.1.1508operator"" "" _ZW() [2/2]	258
5.4.1.1509operator"" "" _zWb() [1/2]	258
5.4.1.1510operator"" "" _zWb() [2/2]	259
5.4.1.1511operator"" "" _ZWb() [1/2]	259
5.4.1.1512operator"" "" _ZWb() [2/2]	259
5.5 oCpt Namespace Reference	259
5.6 oCpt::components Namespace Reference	260
5.7 oCpt::components::comm Namespace Reference	260
5.8 oCpt::components::controller Namespace Reference	260
5.9 oCpt::components::sensors Namespace Reference	260
5.10 oCpt::protocol Namespace Reference	261
5.11 oCpt::vessels Namespace Reference	261

6 Class Documentation	263
6.1 oCpt::Actuator Class Reference	263
6.1.1 Detailed Description	264
6.1.2 Constructor & Destructor Documentation	264
6.1.2.1 Actuator()	264
6.1.2.2 ~Actuator()	264
6.1.3 Member Function Documentation	264
6.1.3.1 run()	264
6.1.3.2 setActuator()	264
6.1.3.3 stop()	265
6.2 oCpt::ActuatorTask Class Reference	265
6.2.1 Detailed Description	266
6.2.2 Constructor & Destructor Documentation	266
6.2.2.1 ActuatorTask()	266
6.2.2.2 ~ActuatorTask()	267
6.3 oCpt::protocol::adc Class Reference	267
6.3.1 Detailed Description	268
6.3.2 Member Typedef Documentation	268
6.3.2.1 ptr	268
6.3.3 Constructor & Destructor Documentation	268
6.3.3.1 adc()	268
6.3.3.2 ~adc()	269
6.3.4 Member Function Documentation	269
6.3.4.1 compare()	269
6.3.4.2 getValue()	270
6.3.4.3 operator==()	270
6.3.5 Member Data Documentation	270
6.3.5.1 device_	270
6.3.5.2 id_	270
6.3.5.3 path_	271

6.3.5.4	value_	271
6.4	oCpt::ARM Class Reference	271
6.4.1	Detailed Description	272
6.4.2	Constructor & Destructor Documentation	272
6.4.2.1	ARM()	272
6.4.2.2	~ARM()	272
6.4.3	Member Function Documentation	273
6.4.3.1	getADC()	273
6.4.3.2	getAdcVector()	273
6.5	oCpt::components::controller::BBB Class Reference	274
6.5.1	Detailed Description	275
6.5.2	Constructor & Destructor Documentation	275
6.5.2.1	BBB()	275
6.5.2.2	~BBB()	275
6.6	oCpt::Boatswain Class Reference	275
6.6.1	Detailed Description	276
6.6.2	Constructor & Destructor Documentation	276
6.6.2.1	Boatswain()	276
6.6.2.2	~Boatswain()	277
6.6.3	Member Function Documentation	277
6.6.3.1	initialize()	277
6.6.3.2	registerActuator()	277
6.6.3.3	registerComm()	277
6.6.3.4	registerSensor()	278
6.6.3.5	resetTimer()	278
6.6.3.6	run()	279
6.6.3.7	stop()	280
6.7	oCpt::Captain Class Reference	280
6.7.1	Detailed Description	281
6.7.2	Constructor & Destructor Documentation	281

6.7.2.1	Captain()	281
6.7.2.2	~Captain()	281
6.7.3	Member Function Documentation	281
6.7.3.1	initialize()	281
6.7.3.2	run()	281
6.7.3.3	stop()	282
6.8	oCpt::CommunicationTask Class Reference	282
6.8.1	Detailed Description	283
6.8.2	Constructor & Destructor Documentation	283
6.8.2.1	CommunicationTask()	283
6.8.2.2	~CommunicationTask()	284
6.9	oCpt::components::sensors::ControlKalmanIMU< T > Class Template Reference	284
6.9.1	Detailed Description	285
6.9.2	Member Typedef Documentation	285
6.9.2.1	Base	285
6.9.3	Constructor & Destructor Documentation	285
6.9.3.1	ControlKalmanIMU() [1/2]	285
6.9.3.2	ControlKalmanIMU() [2/2]	285
6.9.4	Member Function Documentation	286
6.9.4.1	operator=()	286
6.10	oCpt::World::Location::coordinate Struct Reference	286
6.10.1	Detailed Description	286
6.10.2	Member Data Documentation	286
6.10.2.1	direction	286
6.10.2.2	value	286
6.11	oCpt::CoveragePathTask Class Reference	287
6.11.1	Detailed Description	288
6.11.2	Constructor & Destructor Documentation	288
6.11.2.1	CoveragePathTask()	288
6.11.2.2	~CoveragePathTask()	288

6.12 oCpt::DredgeTask Class Reference	289
6.12.1 Detailed Description	290
6.12.2 Constructor & Destructor Documentation	290
6.12.2.1 DredgeTask()	290
6.12.2.2 ~DredgeTask()	290
6.13 oCpt::FollowTask Class Reference	291
6.13.1 Detailed Description	292
6.13.2 Constructor & Destructor Documentation	292
6.13.2.1 FollowTask()	292
6.13.2.2 ~FollowTask()	292
6.14 oCpt::protocol::gpio Class Reference	293
6.14.1 Detailed Description	295
6.14.2 Member Typedef Documentation	295
6.14.2.1 cb_func	295
6.14.2.2 ptr	295
6.14.2.3 signal_t	295
6.14.3 Member Enumeration Documentation	295
6.14.3.1 Direction	295
6.14.3.2 Edge	296
6.14.3.3 Value	296
6.14.4 Constructor & Destructor Documentation	296
6.14.4.1 gpio()	296
6.14.4.2 ~gpio()	297
6.14.5 Member Function Documentation	298
6.14.5.1 exportedGpios()	298
6.14.5.2 exportPin()	298
6.14.5.3 getDirection()	299
6.14.5.4 getEdge()	299
6.14.5.5 getPinNumber()	299
6.14.5.6 getValue()	300

6.14.5.7 <code>internalCbFunc()</code>	300
6.14.5.8 <code>readPinValue() [1/2]</code>	300
6.14.5.9 <code>readPinValue() [2/2]</code>	301
6.14.5.10 <code>setCallbackFunction()</code>	301
6.14.5.11 <code>setDirection()</code>	302
6.14.5.12 <code>setEdge()</code>	302
6.14.5.13 <code>setPinNumber()</code>	302
6.14.5.14 <code>setValue()</code>	303
6.14.5.15 <code>toggle()</code>	303
6.14.5.16 <code>unexportPin()</code>	303
6.14.5.17 <code>waitForEdge()</code>	304
6.14.5.18 <code>waitForEdgeAsync()</code>	304
6.14.5.19 <code>writePinValue() [1/2]</code>	305
6.14.5.20 <code>writePinValue() [2/2]</code>	305
6.14.6 Member Data Documentation	306
6.14.6.1 <code>cb_</code>	306
6.14.6.2 <code>direction_</code>	306
6.14.6.3 <code>edge_</code>	306
6.14.6.4 <code>gpiopath_</code>	306
6.14.6.5 <code>pinNumber_</code>	306
6.14.6.6 <code>signalChanged</code>	307
6.14.6.7 <code>threadRunning_</code>	307
6.14.6.8 <code>value_</code>	307
6.15 <code>oCpt::components::sensors::Gps</code> Class Reference	307
6.15.1 Detailed Description	308
6.15.2 Member Typedef Documentation	309
6.15.2.1 <code>ReturnValue_t</code>	309
6.15.3 Constructor & Destructor Documentation	309
6.15.3.1 <code>Gps()</code>	309
6.15.3.2 <code>~Gps()</code>	309

6.15.4 Member Function Documentation	310
6.15.4.1 interpretMsg()	310
6.15.4.2 run()	310
6.15.4.3 setIoservice()	311
6.15.4.4 stop()	311
6.15.4.5 updateSensor()	312
6.15.5 Member Data Documentation	312
6.15.5.1 device_	312
6.15.5.2 serial_	313
6.16 oCpt::World::Location::gpsPoint Struct Reference	313
6.16.1 Detailed Description	313
6.16.2 Member Function Documentation	314
6.16.2.1 toString()	314
6.16.3 Member Data Documentation	314
6.16.3.1 height	314
6.16.3.2 latitude	314
6.16.3.3 longitude	314
6.17 oCpt::iActuator Class Reference	315
6.17.1 Detailed Description	315
6.17.2 Member Typedef Documentation	315
6.17.2.1 ptr	315
6.17.3 Constructor & Destructor Documentation	316
6.17.3.1 iActuator()	316
6.17.3.2 ~iActuator()	316
6.17.4 Member Function Documentation	316
6.17.4.1 run()	316
6.17.4.2 setActuator()	316
6.17.4.3 stop()	316
6.18 oCpt::iBoatswain Class Reference	317
6.18.1 Detailed Description	318

6.18.2 Member Typedef Documentation	318
6.18.2.1 ptr	318
6.18.2.2 timerPtr	318
6.18.3 Constructor & Destructor Documentation	319
6.18.3.1 iBoatswain()	319
6.18.3.2 ~iBoatswain()	320
6.18.4 Member Function Documentation	320
6.18.4.1 getIoservice()	320
6.18.4.2 getStopThread()	320
6.18.4.3 initialize()	320
6.18.4.4 registerActuator()	320
6.18.4.5 registerComm()	321
6.18.4.6 registerSensor()	321
6.18.4.7 resetTimer()	321
6.18.4.8 run()	322
6.18.4.9 setStopThread()	322
6.18.4.10 stop()	322
6.18.5 Member Data Documentation	322
6.18.5.1 controller_	322
6.18.5.2 ioservice_	323
6.18.5.3 localStopThread_	323
6.18.5.4 manualSensors_	323
6.18.5.5 stopThread_	323
6.18.5.6 timers_	323
6.18.5.7 timerSensors_	323
6.19 oCpt::iCaptain Class Reference	324
6.19.1 Detailed Description	324
6.19.2 Member Typedef Documentation	324
6.19.2.1 ptr	324
6.19.3 Constructor & Destructor Documentation	325

6.19.3.1	iCaptain()	325
6.19.3.2	~iCaptain()	325
6.19.4	Member Function Documentation	325
6.19.4.1	getStopThread_()	325
6.19.4.2	initialize()	325
6.19.4.3	run()	325
6.19.4.4	setStopThread_()	325
6.19.4.5	stop()	326
6.19.5	Member Data Documentation	326
6.19.5.1	localStopThread_	326
6.19.5.2	stopThread_	326
6.19.5.3	world_	326
6.20	oCpt::iComm Class Reference	326
6.20.1	Detailed Description	327
6.20.2	Member Typedef Documentation	328
6.20.2.1	ptr	328
6.20.2.2	signal_t	328
6.20.3	Constructor & Destructor Documentation	328
6.20.3.1	iComm()	328
6.20.3.2	~iComm()	328
6.20.4	Member Function Documentation	328
6.20.4.1	getId()	328
6.20.4.2	getMsgQueue()	329
6.20.4.3	getTypeOfComm()	329
6.20.4.4	initialize()	329
6.20.4.5	readFiFoMsg()	330
6.20.4.6	recieveAsyncMessage()	330
6.20.4.7	recieveMessage()	330
6.20.4.8	run()	330
6.20.4.9	sendMessage()	330

6.20.4.10 setId()	331
6.20.4.11 setIoservice()	331
6.20.4.12 setTypeOfComm()	331
6.20.4.13 stop()	332
6.20.5 Member Data Documentation	332
6.20.5.1 device_	332
6.20.5.2 id_	332
6.20.5.3 ioservice_	332
6.20.5.4 msgQueue_	332
6.20.5.5 msgRecievedSig	333
6.20.5.6 timer_	333
6.20.5.7 typeOfComm_	333
6.20.5.8 world_	333
6.21 oCpt::iController Class Reference	333
6.21.1 Detailed Description	334
6.21.2 Member Typedef Documentation	334
6.21.2.1 io_t	334
6.21.2.2 ptr	334
6.21.3 Constructor & Destructor Documentation	334
6.21.3.1 iController()	334
6.21.3.2 ~iController()	335
6.21.4 Member Function Documentation	335
6.21.4.1 getADC()	335
6.21.4.2 getAdcVector()	335
6.21.5 Member Data Documentation	336
6.21.5.1 adcVector_	336
6.21.5.2 world_	336
6.22 oCpt::iSensor Class Reference	336
6.22.1 Detailed Description	338
6.22.2 Member Typedef Documentation	338

6.22.2.1	generic_t	338
6.22.2.2	ptr	338
6.22.2.3	signal_t	338
6.22.3	Constructor & Destructor Documentation	338
6.22.3.1	iSensor()	338
6.22.3.2	~iSensor()	339
6.22.4	Member Function Documentation	339
6.22.4.1	getID()	339
6.22.4.2	getSig()	339
6.22.4.3	getState()	340
6.22.4.4	getTimer()	340
6.22.4.5	getTypeOfSensor()	340
6.22.4.6	init()	340
6.22.4.7	operator==()	340
6.22.4.8	run()	341
6.22.4.9	setID()	341
6.22.4.10	setIoservice()	342
6.22.4.11	setTimer()	342
6.22.4.12	setTypeOfSensor()	342
6.22.4.13	stop()	342
6.22.4.14	updateSensor()	343
6.22.5	Member Data Documentation	343
6.22.5.1	controller_	343
6.22.5.2	id_	343
6.22.5.3	ioservice_	343
6.22.5.4	sensorRunning_	343
6.22.5.5	sig_	344
6.22.5.6	state_	344
6.22.5.7	timer_	344
6.22.5.8	typeOfSensor_	344

6.22.5.9 <code>world_</code>	344
6.23 <code>oCpt::iTask</code> Class Reference	345
6.23.1 Detailed Description	346
6.23.2 Member Typedef Documentation	346
6.23.2.1 <code>ptr</code>	346
6.23.2.2 <code>taskqueue</code>	346
6.23.3 Member Enumeration Documentation	346
6.23.3.1 <code>TypeOf</code>	346
6.23.4 Constructor & Destructor Documentation	347
6.23.4.1 <code>iTask()</code>	347
6.23.4.2 <code>~iTask()</code>	347
6.23.5 Member Function Documentation	347
6.23.5.1 <code>start()</code>	347
6.23.5.2 <code>status()</code>	347
6.23.5.3 <code>stop()</code>	348
6.23.6 Member Data Documentation	348
6.23.6.1 <code>_concurrent</code>	348
6.23.6.2 <code>_vessel</code>	348
6.23.6.3 <code>Work</code>	348
6.24 <code>oCpt::iVessel</code> Class Reference	349
6.24.1 Detailed Description	349
6.24.2 Member Typedef Documentation	350
6.24.2.1 <code>ptr</code>	350
6.24.3 Constructor & Destructor Documentation	350
6.24.3.1 <code>iVessel() [1/2]</code>	350
6.24.3.2 <code>iVessel() [2/2]</code>	350
6.24.3.3 <code>~iVessel()</code>	350
6.24.4 Member Function Documentation	351
6.24.4.1 <code>getStopThread()</code>	351
6.24.4.2 <code>initialize()</code>	351

6.24.4.3	run()	351
6.24.4.4	setStopThread()	351
6.24.4.5	stop()	352
6.24.5	Member Data Documentation	352
6.24.5.1	stopThread_	352
6.25	oCpt::components::sensors::KalmanIMU Class Reference	352
6.25.1	Detailed Description	354
6.25.2	Member Typedef Documentation	354
6.25.2.1	ReturnValue_t	354
6.25.3	Constructor & Destructor Documentation	354
6.25.3.1	KalmanIMU()	354
6.25.3.2	~KalmanIMU()	354
6.25.4	Member Function Documentation	355
6.25.4.1	init()	355
6.25.4.2	RazorUpdate()	355
6.25.4.3	run()	356
6.25.4.4	setIOService()	356
6.25.4.5	updateSensor()	357
6.25.5	Member Data Documentation	357
6.25.5.1	device_	357
6.25.5.2	razor_	357
6.26	oCpt::World::Location Class Reference	357
6.26.1	Detailed Description	358
6.26.2	Member Typedef Documentation	358
6.26.2.1	coordinate_t	358
6.26.2.2	degree_t	358
6.26.2.3	gpsPoint_t	359
6.26.2.4	ptr	359
6.26.3	Member Enumeration Documentation	359
6.26.3.1	cardinal_direction	359

6.26.4 Constructor & Destructor Documentation	359
6.26.4.1 Location()	359
6.26.4.2 ~Location()	359
6.26.5 Member Function Documentation	359
6.26.5.1 getCurrentLocation()	359
6.26.5.2 getLocationHistory()	360
6.26.5.3 push_back()	360
6.26.5.4 stocd()	360
6.26.6 Member Data Documentation	361
6.26.6.1 currentLocation_	361
6.26.6.2 LocationHistory	361
6.27 oCpt::World::Time< T > Class Template Reference	361
6.27.1 Detailed Description	362
6.27.2 Member Typedef Documentation	362
6.27.2.1 ptr	362
6.27.3 Constructor & Destructor Documentation	362
6.27.3.1 Log() [1/2]	362
6.27.3.2 Log() [2/2]	363
6.27.3.3 ~Log()	364
6.27.4 Member Function Documentation	364
6.27.4.1 getEpoch()	364
6.27.4.2 getValue()	364
6.27.5 Member Data Documentation	365
6.27.5.1 _epoch	365
6.27.5.2 _value	365
6.28 oCpt::LogTask Class Reference	365
6.28.1 Detailed Description	366
6.28.2 Constructor & Destructor Documentation	366
6.28.2.1 LogTask()	366
6.28.2.2 ~LogTask()	367

6.29 oCpt::LoRa Class Reference	367
6.29.1 Detailed Description	370
6.29.2 Member Enumeration Documentation	370
6.29.2.1 BandWidth	370
6.29.2.2 CodingRate	371
6.29.2.3 GetSet	371
6.29.2.4 MacCommand	371
6.29.2.5 ModulationMode	372
6.29.2.6 RadioBandWidth	372
6.29.2.7 RadioCommand	372
6.29.2.8 SpreadingFactor	373
6.29.3 Constructor & Destructor Documentation	373
6.29.3.1 LoRa()	373
6.29.3.2 ~LoRa()	374
6.29.4 Member Function Documentation	374
6.29.4.1 bandWidthToString()	374
6.29.4.2 buildMacCmdString()	374
6.29.4.3 buildRadioCmdString() [1/2]	374
6.29.4.4 buildRadioCmdString() [2/2]	375
6.29.4.5 calculateDownTime()	375
6.29.4.6 codingRateToString()	376
6.29.4.7 encodeTypeToHex()	376
6.29.4.8 hexToString()	376
6.29.4.9 initialize()	377
6.29.4.10 macpause()	377
6.29.4.11 messageRecieved()	378
6.29.4.12 recieveAsyncMessage()	379
6.29.4.13 recieveMessage()	379
6.29.4.14 run()	379
6.29.4.15 rx()	380

6.29.4.16 sendMessage()	380
6.29.4.17 stop()	381
6.29.4.18 stringToHex()	381
6.29.4.19 write()	381
6.29.5 Member Data Documentation	382
6.29.5.1 afcbw_	382
6.29.5.2 baudrate_	382
6.29.5.3 bw_	382
6.29.5.4 cr_	383
6.29.5.5 crc_	383
6.29.5.6 fdev_	383
6.29.5.7 freq_	383
6.29.5.8 fskBitRate_	383
6.29.5.9 ignoreWarn_	384
6.29.5.10 listen_	384
6.29.5.11 mod_	384
6.29.5.12 prlen_	384
6.29.5.13 proceed_	384
6.29.5.14 pwr_	384
6.29.5.15 rxbw_	385
6.29.5.16 sendAllowed_	385
6.29.5.17 serial_	385
6.29.5.18 sf_	385
6.29.5.19 sync_	385
6.29.5.20 wdt_	386
6.30 oCpt::components::comm::LoRa_RN2483 Class Reference	386
6.30.1 Detailed Description	387
6.30.2 Constructor & Destructor Documentation	387
6.30.2.1 LoRa_RN2483()	387
6.30.2.2 ~LoRa_RN2483()	388

6.31 oCpt::vessels::Meetcatamaran Class Reference	388
6.31.1 Detailed Description	389
6.31.2 Constructor & Destructor Documentation	389
6.31.2.1 Meetcatamaran()	389
6.31.2.2 ~Meetcatamaran()	389
6.32 oCpt::iComm::Message Struct Reference	389
6.32.1 Detailed Description	390
6.32.2 Member Typedef Documentation	390
6.32.2.1 ptr	390
6.32.3 Constructor & Destructor Documentation	390
6.32.3.1 Message()	390
6.32.3.2 ~Message()	390
6.32.4 Member Data Documentation	391
6.32.4.1 Payload	391
6.32.4.2 Stamp	391
6.33 oCpt::oCptException Class Reference	391
6.33.1 Detailed Description	392
6.33.2 Constructor & Destructor Documentation	392
6.33.2.1 oCptException()	392
6.33.2.2 ~oCptException()	392
6.33.3 Member Function Documentation	392
6.33.3.1 what()	392
6.33.4 Member Data Documentation	392
6.33.4.1 _id	392
6.33.4.2 _msg	393
6.34 oCpt::components::sensors::OrientationMeasurementKalmanIMU< T > Class Template Reference	393
6.34.1 Detailed Description	394
6.34.2 Member Typedef Documentation	394
6.34.2.1 Base	394
6.34.3 Constructor & Destructor Documentation	395

6.34.3.1	OrientationMeasurementKalmanIMU() [1/2]	395
6.34.3.2	OrientationMeasurementKalmanIMU() [2/2]	395
6.34.4	Member Function Documentation	395
6.34.4.1	operator=()	395
6.34.4.2	phi() [1/2]	395
6.34.4.3	phi() [2/2]	396
6.34.4.4	phiPrime() [1/2]	396
6.34.4.5	phiPrime() [2/2]	396
6.34.4.6	psi() [1/2]	397
6.34.4.7	psi() [2/2]	397
6.34.4.8	psiPrime() [1/2]	397
6.34.4.9	psiPrime() [2/2]	398
6.34.4.10	theta() [1/2]	398
6.34.4.11	theta() [2/2]	398
6.34.4.12	thetaPrime() [1/2]	399
6.34.4.13	thetaPrime() [2/2]	399
6.34.5	Member Data Documentation	399
6.34.5.1	Phi	399
6.34.5.2	PhiPrime	399
6.34.5.3	Psi	400
6.34.5.4	PsiPrime	400
6.34.5.5	Theta	400
6.34.5.6	ThetaPrime	400
6.35	oCpt::components::sensors::OrientationMeasurementModelKalmanIMU< T, CovarianceBase > Class Template Reference	401
6.35.1	Detailed Description	401
6.35.2	Member Typedef Documentation	402
6.35.2.1	M	402
6.35.2.2	S	402
6.35.3	Constructor & Destructor Documentation	402
6.35.3.1	OrientationMeasurementModelKalmanIMU()	402

6.35.4 Member Function Documentation	402
6.35.4.1 h()	402
6.36 oCpt::PathTask Class Reference	403
6.36.1 Detailed Description	405
6.36.2 Constructor & Destructor Documentation	405
6.36.2.1 PathTask()	405
6.36.2.2 ~PathTask()	405
6.37 oCpt::components::sensors::PositionMeasurementKalmanIMU< T > Class Template Reference	406
6.37.1 Detailed Description	407
6.37.2 Member Typedef Documentation	407
6.37.2.1 Base	407
6.37.3 Constructor & Destructor Documentation	407
6.37.3.1 PositionMeasurementKalmanIMU() [1/2]	407
6.37.3.2 PositionMeasurementKalmanIMU() [2/2]	407
6.37.4 Member Function Documentation	408
6.37.4.1 acc_x() [1/2]	408
6.37.4.2 acc_x() [2/2]	408
6.37.4.3 acc_y() [1/2]	408
6.37.4.4 acc_y() [2/2]	409
6.37.4.5 acc_z() [1/2]	409
6.37.4.6 acc_z() [2/2]	409
6.37.4.7 operator=()	409
6.37.5 Member Data Documentation	410
6.37.5.1 accX	410
6.37.5.2 accY	410
6.37.5.3 accZ	410
6.38 oCpt::components::sensors::PositionMeasurementModelKalmanIMU< T, CovarianceBase > Class Template Reference	410
6.38.1 Detailed Description	411
6.38.2 Member Typedef Documentation	411
6.38.2.1 M	411

6.38.2.2	S	411
6.38.3	Constructor & Destructor Documentation	411
6.38.3.1	PositionMeasurementModelKalmanIMU()	411
6.38.4	Member Function Documentation	412
6.38.4.1	h()	412
6.39	oCpt::components::sensors::PT100 Class Reference	412
6.39.1	Detailed Description	414
6.39.2	Member Typedef Documentation	414
6.39.2.1	ReturnValue_t	414
6.39.3	Constructor & Destructor Documentation	414
6.39.3.1	PT100()	414
6.39.3.2	~PT100()	415
6.39.4	Member Function Documentation	415
6.39.4.1	init()	415
6.39.4.2	run()	415
6.39.4.3	setCalibrationTemperature()	416
6.39.4.4	stop()	416
6.39.4.5	updateSensor()	416
6.39.5	Member Data Documentation	417
6.39.5.1	_analogeValue	417
6.39.5.2	_constant	417
6.39.5.3	_device	417
6.39.5.4	_dy_dx	417
6.39.5.5	_pinid	417
6.40	oCpt::components::sensors::Razor Class Reference	418
6.40.1	Detailed Description	419
6.40.2	Member Typedef Documentation	419
6.40.2.1	Acceleration_t	419
6.40.2.2	Angular_velocity_t	420
6.40.2.3	Magnetic_flux_density_t	420

6.40.2.4	ReturnValue_t	420
6.40.3	Member Enumeration Documentation	420
6.40.3.1	Mode	420
6.40.4	Constructor & Destructor Documentation	420
6.40.4.1	Razor()	420
6.40.4.2	~Razor()	421
6.40.5	Member Function Documentation	421
6.40.5.1	checkLRC()	421
6.40.5.2	fillReturnValue()	421
6.40.5.3	getFreq()	422
6.40.5.4	getMode()	422
6.40.5.5	init()	422
6.40.5.6	msgHandler()	423
6.40.5.7	run()	423
6.40.5.8	setFreq()	424
6.40.5.9	setIoservice()	424
6.40.5.10	setMode()	425
6.40.5.11	stop()	425
6.40.5.12	updateSensor()	426
6.40.6	Member Data Documentation	426
6.40.6.1	cb	426
6.40.6.2	device_	426
6.40.6.3	freq_	426
6.40.6.4	mode_	427
6.40.6.5	serial_	427
6.41	oCpt::components::sensors::Razor::ReturnValue Struct Reference	427
6.41.1	Detailed Description	427
6.41.2	Member Data Documentation	427
6.41.2.1	acc	427
6.41.2.2	gyro	428

6.41.2.3	mag	428
6.42	oCpt::components::sensors::KalmanIMU::ReturnValue Struct Reference	428
6.42.1	Detailed Description	428
6.42.2	Member Data Documentation	428
6.42.2.1	acceleration	428
6.42.2.2	orientation	429
6.42.2.3	orientation_change	429
6.42.2.4	position	429
6.42.2.5	velocity	429
6.43	oCpt::World::Location::RoutePoint Struct Reference	429
6.43.1	Detailed Description	430
6.43.2	Member Typedef Documentation	430
6.43.2.1	ptr	430
6.43.3	Member Data Documentation	430
6.43.3.1	Location	430
6.43.3.2	TimePoint	430
6.44	oCpt::RouteTask Class Reference	431
6.44.1	Detailed Description	432
6.44.2	Constructor & Destructor Documentation	432
6.44.2.1	RouteTask()	432
6.44.2.2	~RouteTask()	432
6.45	oCpt::Sensor Class Reference	433
6.45.1	Detailed Description	434
6.45.2	Constructor & Destructor Documentation	434
6.45.2.1	Sensor()	434
6.45.2.2	~Sensor()	434
6.45.3	Member Function Documentation	434
6.45.3.1	init()	434
6.45.3.2	run()	435
6.45.3.3	setIoservice()	436

6.45.3.4	stop()	436
6.45.3.5	updateSensor()	437
6.46	oCpt::SensorTask Class Reference	438
6.46.1	Detailed Description	439
6.46.2	Constructor & Destructor Documentation	439
6.46.2.1	SensorTask()	439
6.46.2.2	~SensorTask()	440
6.47	oCpt::protocol::Serial Class Reference	440
6.47.1	Detailed Description	442
6.47.2	Member Typedef Documentation	442
6.47.2.1	cb_func	442
6.47.2.2	character_size_t	442
6.47.2.3	flow_control_t	442
6.47.2.4	io_service_t	443
6.47.2.5	parity_t	443
6.47.2.6	ptr	443
6.47.2.7	serialport_t	443
6.47.2.8	signal_t	443
6.47.2.9	stop_bits_t	443
6.47.3	Constructor & Destructor Documentation	443
6.47.3.1	Serial()	443
6.47.4	Member Function Documentation	444
6.47.4.1	close()	444
6.47.4.2	closeCallback()	445
6.47.4.3	getReturnMsgQueue()	446
6.47.4.4	internalCallback()	446
6.47.4.5	isOpen()	447
6.47.4.6	open()	448
6.47.4.7	readComplete()	448
6.47.4.8	readFiFoMsg()	449

6.47.4.9 <code>ReadStart()</code>	450
6.47.4.10 <code>setIoservice()</code>	450
6.47.4.11 <code>setMaxReadLength()</code>	451
6.47.4.12 <code>setReadCallback()</code>	451
6.47.4.13 <code>start()</code>	451
6.47.4.14 <code>write() [1/2]</code>	452
6.47.4.15 <code>write() [2/2]</code>	452
6.47.4.16 <code>writeCallback()</code>	453
6.47.4.17 <code>writeComplete()</code>	454
6.47.4.18 <code>writeStart()</code>	455
6.47.5 Member Data Documentation	455
6.47.5.1 <code>baudrate_</code>	455
6.47.5.2 <code>callback_</code>	455
6.47.5.3 <code>csize_</code>	456
6.47.5.4 <code>device_</code>	456
6.47.5.5 <code>firstMsg</code>	456
6.47.5.6 <code>flow_</code>	456
6.47.5.7 <code>ioservice_</code>	456
6.47.5.8 <code>maxReadLength_</code>	456
6.47.5.9 <code>msg_</code>	457
6.47.5.10 <code>msgQueue_</code>	457
6.47.5.11 <code>msgRecievedSig</code>	457
6.47.5.12 <code>parity_</code>	457
6.47.5.13 <code>read_msg</code>	457
6.47.5.14 <code>receivedMsg_</code>	457
6.47.5.15 <code>returnMsgQueue_</code>	458
6.47.5.16 <code>serialport_</code>	458
6.47.5.17 <code>stop_</code>	458
6.48 <code>oCpt::iSensor::State</code> Struct Reference	458
6.48.1 Detailed Description	458

6.48.2 Member Data Documentation	459
6.48.2.1 Stamp	459
6.48.2.2 Value	459
6.49 oCpt::components::sensors::StateKalmanIMU< T > Class Template Reference	459
6.49.1 Detailed Description	461
6.49.2 Member Typedef Documentation	461
6.49.2.1 Base	461
6.49.3 Constructor & Destructor Documentation	461
6.49.3.1 StateKalmanIMU() [1/2]	461
6.49.3.2 StateKalmanIMU() [2/2]	462
6.49.4 Member Function Documentation	462
6.49.4.1 acc_x() [1/2]	462
6.49.4.2 acc_x() [2/2]	462
6.49.4.3 acc_y() [1/2]	463
6.49.4.4 acc_y() [2/2]	463
6.49.4.5 acc_z() [1/2]	463
6.49.4.6 acc_z() [2/2]	464
6.49.4.7 operator=()	464
6.49.4.8 phi() [1/2]	464
6.49.4.9 phi() [2/2]	464
6.49.4.10 phiPrime() [1/2]	465
6.49.4.11 phiPrime() [2/2]	465
6.49.4.12 pos_x() [1/2]	465
6.49.4.13 pos_x() [2/2]	465
6.49.4.14 pos_y() [1/2]	466
6.49.4.15 pos_y() [2/2]	466
6.49.4.16 pos_z() [1/2]	466
6.49.4.17 pos_z() [2/2]	466
6.49.4.18 psi() [1/2]	466
6.49.4.19 psi() [2/2]	467

6.49.4.20 psiPrime() [1/2]	467
6.49.4.21 psiPrime() [2/2]	467
6.49.4.22 theta() [1/2]	468
6.49.4.23 theta() [2/2]	468
6.49.4.24 thetaPrime() [1/2]	468
6.49.4.25 thetaPrime() [2/2]	469
6.49.4.26 vel_x() [1/2]	469
6.49.4.27 vel_x() [2/2]	469
6.49.4.28 vel_y() [1/2]	469
6.49.4.29 vel_y() [2/2]	469
6.49.4.30 vel_z() [1/2]	469
6.49.4.31 vel_z() [2/2]	470
6.49.5 Member Data Documentation	470
6.49.5.1 accX	470
6.49.5.2 accY	470
6.49.5.3 accZ	470
6.49.5.4 Phi	470
6.49.5.5 PhiPrime	471
6.49.5.6 posX	471
6.49.5.7 posY	471
6.49.5.8 posZ	471
6.49.5.9 Psi	471
6.49.5.10 PsiPrime	472
6.49.5.11 Theta	472
6.49.5.12 ThetaPrime	472
6.49.5.13 velX	472
6.49.5.14 velY	472
6.49.5.15 velZ	473
6.50 oCpt::iTask::Status Class Reference	473
6.50.1 Detailed Description	473

6.50.2 Member Typedef Documentation	473
6.50.2.1 ptr	473
6.50.3 Constructor & Destructor Documentation	474
6.50.3.1 Status()	474
6.50.3.2 ~Status()	474
6.50.4 Member Function Documentation	474
6.50.4.1 progress()	474
6.50.4.2 running()	474
6.50.4.3 successful()	475
6.50.5 Member Data Documentation	475
6.50.5.1 __progress	475
6.50.5.2 __running	475
6.50.5.3 __successful	475
6.51 oCpt::components::sensors::SystemModelKalmanIMU< T, CovarianceBase > Class Template Reference	476
6.51.1 Detailed Description	476
6.51.2 Member Typedef Documentation	477
6.51.2.1 C	477
6.51.2.2 S	477
6.51.3 Member Function Documentation	477
6.51.3.1 f()	477
6.52 oCpt::Task Class Reference	477
6.52.1 Detailed Description	478
6.52.2 Constructor & Destructor Documentation	478
6.52.2.1 Task()	478
6.52.2.2 ~Task()	479
6.52.3 Member Function Documentation	479
6.52.3.1 start()	479
6.52.3.2 status()	479
6.52.3.3 stop()	479
6.52.4 Member Data Documentation	480

6.52.4.1 <code>_status</code>	480
6.52.4.2 <code>_typeof</code>	480
6.53 <code>oCpt::World::Time</code> Class Reference	480
6.53.1 Detailed Description	481
6.53.2 Member Typedef Documentation	481
6.53.2.1 <code>clock_t</code>	481
6.53.2.2 History	481
6.53.2.3 <code>ptr</code>	481
6.53.2.4 <code>tick_period</code>	481
6.53.2.5 <code>timepoint_t</code>	482
6.53.3 Constructor & Destructor Documentation	482
6.53.3.1 <code>Time()</code>	482
6.53.3.2 <code>~Time()</code>	482
6.53.4 Member Function Documentation	482
6.53.4.1 <code>getTimeClock()</code>	482
6.53.4.2 <code>now()</code>	483
6.53.5 Member Data Documentation	483
6.53.5.1 <code>timeClock_</code>	483
6.54 <code>oCpt::protocol::userspace</code> Class Reference	483
6.54.1 Detailed Description	484
6.54.2 Constructor & Destructor Documentation	484
6.54.2.1 <code>userspace()</code>	484
6.54.2.2 <code>~userspace()</code>	484
6.54.3 Member Function Documentation	484
6.54.3.1 <code>dtboLoaded()</code>	484
6.54.3.2 <code>fileExist()</code>	485
6.54.3.3 <code>modLoaded()</code>	485
6.54.4 Member Data Documentation	486
6.54.4.1 <code>usMutex</code>	486
6.55 <code>oCpt::Vessel</code> Class Reference	487

6.55.1	Detailed Description	488
6.55.2	Constructor & Destructor Documentation	488
6.55.2.1	Vessel() [1/2]	488
6.55.2.2	Vessel() [2/2]	488
6.55.2.3	~Vessel()	489
6.55.3	Member Function Documentation	489
6.55.3.1	initialize()	489
6.55.3.2	run()	489
6.55.3.3	stop()	490
6.55.4	Member Data Documentation	490
6.55.4.1	actuators_	490
6.55.4.2	boatswain_	490
6.55.4.3	captain_	490
6.55.4.4	comm_	490
6.55.4.5	controller_	491
6.55.4.6	sensors_	491
6.55.4.7	world_	491
6.56	oCpt::WorkTask Class Reference	491
6.56.1	Detailed Description	492
6.56.2	Constructor & Destructor Documentation	492
6.56.2.1	WorkTask()	492
6.56.2.2	~WorkTask()	493
6.57	oCpt::World Class Reference	493
6.57.1	Detailed Description	494
6.57.2	Member Typedef Documentation	494
6.57.2.1	ptr	494
6.57.3	Constructor & Destructor Documentation	494
6.57.3.1	World()	494
6.57.3.2	~World()	494
6.57.4	Member Function Documentation	494
6.57.4.1	getTime()	494
6.57.4.2	now()	495
6.57.5	Member Data Documentation	495
6.57.5.1	time_	495

7 File Documentation	497
7.1 /projects/mti/ohCaptain/ohCaptain/include/Communication/LoRa_RN2483.h File Reference	497
7.2 /projects/mti/ohCaptain/ohCaptain/include/Controllers/BeagleboneBlack.h File Reference	498
7.3 /projects/mti/ohCaptain/ohCaptain/include/Core/Actuator.h File Reference	499
7.4 /projects/mti/ohCaptain/ohCaptain/include/Core/Boatswain.h File Reference	499
7.5 /projects/mti/ohCaptain/ohCaptain/include/Core/Captain.h File Reference	500
7.6 /projects/mti/ohCaptain/ohCaptain/include/Core/Communication.h File Reference	501
7.7 /projects/mti/ohCaptain/ohCaptain/include/Core/constants.h File Reference	502
7.8 /projects/mti/ohCaptain/ohCaptain/include/Core/Controller.h File Reference	504
7.8.1 Macro Definition Documentation	505
7.8.1.1 ADC_IO_BASE_PATH	505
7.8.1.2 ADC_VOLTAGE_PATH	505
7.8.1.3 ADC_VOLTAGE_SUB_PATH	505
7.8.1.4 BBB_CAPE_Mngr	505
7.8.1.5 GPIO_BASE_PATH	505
7.8.1.6 MAX_READ_LENGTH	506
7.8.1.7 MODULE_PATH	506
7.9 /projects/mti/ohCaptain/ohCaptain/include/Core/Exception.h File Reference	506
7.10 /projects/mti/ohCaptain/ohCaptain/include/Core/literals.h File Reference	507
7.10.1 Macro Definition Documentation	538
7.10.1.1 BOOST_UNITS_LITERAL	538
7.10.1.2 BOOST_UNITS_LITERAL_SET	538
7.11 /projects/mti/ohCaptain/ohCaptain/include/Core/Sensor.h File Reference	539
7.11.1 Macro Definition Documentation	540
7.11.1.1 CAST	540
7.12 /projects/mti/ohCaptain/ohCaptain/include/Core/Task.h File Reference	540
7.13 /projects/mti/ohCaptain/ohCaptain/include/Core/Vessel.h File Reference	541
7.14 /projects/mti/ohCaptain/ohCaptain/include/Core/World.h File Reference	542
7.15 /projects/mti/ohCaptain/ohCaptain/include/Sensors/Gps.h File Reference	543
7.16 /projects/mti/ohCaptain/ohCaptain/include/Sensors/KalmanIMU.h File Reference	544

7.17 /projects/mti/ohCaptain/ohCaptain/include/Sensors/PT100.h File Reference	545
7.18 /projects/mti/ohCaptain/ohCaptain/include/Sensors/Razor.h File Reference	546
7.19 /projects/mti/ohCaptain/ohCaptain/include/Vessels/Meetcatamaran.h File Reference	547
7.20 /projects/mti/ohCaptain/ohCaptain/src/Communication/LoRa_RN2483.cpp File Reference	548
7.21 /projects/mti/ohCaptain/ohCaptain/src/Controllers/BeagleboneBlack.cpp File Reference	549
7.22 /projects/mti/ohCaptain/ohCaptain/src/Core/Actuator.cpp File Reference	549
7.23 /projects/mti/ohCaptain/ohCaptain/src/Core/Boatswain.cpp File Reference	549
7.24 /projects/mti/ohCaptain/ohCaptain/src/Core/Captain.cpp File Reference	550
7.25 /projects/mti/ohCaptain/ohCaptain/src/Core/Communication.cpp File Reference	550
7.26 /projects/mti/ohCaptain/ohCaptain/src/Core/Controller.cpp File Reference	551
7.27 /projects/mti/ohCaptain/ohCaptain/src/Core/Sensor.cpp File Reference	551
7.28 /projects/mti/ohCaptain/ohCaptain/src/Core/Task.cpp File Reference	551
7.29 /projects/mti/ohCaptain/ohCaptain/src/Core/Vessel.cpp File Reference	552
7.30 /projects/mti/ohCaptain/ohCaptain/src/Core/World.cpp File Reference	552
7.31 /projects/mti/ohCaptain/ohCaptain/src/Sensors/Gps.cpp File Reference	553
7.32 /projects/mti/ohCaptain/ohCaptain/src/Sensors/KalmanIMU.cpp File Reference	553
7.33 /projects/mti/ohCaptain/ohCaptain/src/Sensors/PT100.cpp File Reference	553
7.34 /projects/mti/ohCaptain/ohCaptain/src/Sensors/Razor.cpp File Reference	554
7.35 /projects/mti/ohCaptain/ohCaptain/src/Vessels/Meetcatamaran.cpp File Reference	554

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

boost	9
boost::units	9
boost::units::constants	9
boost::units::literals	15
oCpt	259
oCpt::components	260
oCpt::components::comm	260
oCpt::components::controller	260
oCpt::components::sensors	260
oCpt::protocol	261
oCpt::vessels	261

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

oCpt::World::Location::coordinate	286
enable_shared_from_this	
oCpt::iBoatswain	317
oCpt::Boatswain	275
exception	
oCpt::oCptException	391
oCpt::World::Location::gpsPoint	313
oCpt::iActuator	315
oCpt::Actuator	263
oCpt::iCaptain	324
oCpt::Captain	280
oCpt::iComm	326
oCpt::LoRa	367
oCpt::components::comm::LoRa_RN2483	386
oCpt::iController	333
oCpt::ARM	271
oCpt::components::controller::BBB	274
oCpt::iSensor	336
oCpt::Sensor	433
oCpt::components::sensors::Gps	307
oCpt::components::sensors::KalmanIMU	352
oCpt::components::sensors::PT100	412
oCpt::components::sensors::Razor	418
oCpt::iTak	345
oCpt::Task	477
oCpt::RouteTask	431
oCpt::CoveragePathTask	287
oCpt::FollowTask	291
oCpt::PathTask	403
oCpt::WorkTask	491
oCpt::ActuatorTask	265
oCpt::CommunicationTask	282
oCpt::DredgeTask	289

oCpt::LogTask	365
oCpt::SensorTask	438
oCpt::iVessel	349
oCpt::Vessel	487
oCpt::vessels::Meetcatamaran	388
LinearizedMeasurementModel	
oCpt::components::sensors::OrientationMeasurementModelKalmanIMU< T, CovarianceBase >	401
oCpt::components::sensors::PositionMeasurementModelKalmanIMU< T, CovarianceBase >	410
LinearizedSystemModel	
oCpt::components::sensors::SystemModelKalmanIMU< T, CovarianceBase >	476
oCpt::World::Location	357
oCpt::World::Time::Log< T >	361
oCpt::iComm::Message	389
oCpt::components::sensors::Razor::ReturnValue	427
oCpt::components::sensors::KalmanIMU::ReturnValue	428
oCpt::World::Location::RoutePoint	429
oCpt::iSensor::State	458
oCpt::iTTask::Status	473
oCpt::World::Time	480
oCpt::protocol::userspace	483
oCpt::protocol::adc	267
oCpt::protocol::gpio	293
oCpt::protocol::Serial	440
Vector	
oCpt::components::sensors::ControlKalmanIMU< T >	284
oCpt::components::sensors::OrientationMeasurementKalmanIMU< T >	393
oCpt::components::sensors::PositionMeasurementKalmanIMU< T >	406
oCpt::components::sensors::StateKalmanIMU< T >	459
oCpt::World	493

Chapter 3

Class Index

3.1 Class List

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

oCpt::Actuator	263
oCpt::ActuatorTask	265
oCpt::protocol::adc	267
oCpt::ARM	271
oCpt::components::controller::BBB	274
oCpt::Boatswain	275
oCpt::Captain	280
oCpt::CommunicationTask	282
oCpt::components::sensors::ControlKalmanIMU< T >	284
oCpt::World::Location::coordinate	286
oCpt::CoveragePathTask	
An object representing a coverage path task	287
oCpt::DredgeTask	
An Object representing a dredging task	289
oCpt::FollowTask	
An object representing a follow the target task	291
oCpt::protocol::gpio	293
oCpt::components::sensors::Gps	307
oCpt::World::Location::gpsPoint	313
oCpt::iActuator	315
oCpt::iBoatswain	317
oCpt::iCaptain	324
oCpt::iComm	326
oCpt::iController	333
oCpt::iSensor	336
oCpt::iTak	
Task interface, all tasks need to adhere to this structure	345
oCpt::iVessel	349
oCpt::components::sensors::KalmanIMU	352
oCpt::World::Location	357
oCpt::World::Time::Log< T >	361
oCpt::LogTask	
An Object representing a data logging task	365
oCpt::LoRa	367
oCpt::components::comm::LoRa_RN2483	386

oCpt::vessels::Meetcatamaran	388
oCpt::iComm::Message	389
oCpt::oCptException	391
oCpt::components::sensors::OrientationMeasurementKalmanIMU< T >	393
oCpt::components::sensors::OrientationMeasurementModelKalmanIMU< T, CovarianceBase >	401
oCpt::PathTask	
An object representing a normal A to B type of path planning	403
oCpt::components::sensors::PositionMeasurementKalmanIMU< T >	406
oCpt::components::sensors::PositionMeasurementModelKalmanIMU< T, CovarianceBase >	410
oCpt::components::sensors::PT100	412
oCpt::components::sensors::Razor	418
oCpt::components::sensors::Razor::ReturnValue	427
oCpt::components::sensors::KalmanIMU::ReturnValue	428
oCpt::World::Location::RoutePoint	429
oCpt::RouteTask	431
oCpt::Sensor	433
oCpt::SensorTask	438
oCpt::protocol::Serial	440
oCpt::ISensor::State	458
oCpt::components::sensors::StateKalmanIMU< T >	459
oCpt::iTask::Status	473
oCpt::components::sensors::SystemModelKalmanIMU< T, CovarianceBase >	476
oCpt::Task	477
oCpt::World::Time	480
oCpt::protocol::userspace	483
oCpt::Vessel	487
oCpt::WorkTask	491
oCpt::World	493

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

/projects/mti/ohCaptain/ohCaptain/include/Communication/ LoRa_RN2483.h	497
/projects/mti/ohCaptain/ohCaptain/include/Controllers/ BeagleboneBlack.h	498
/projects/mti/ohCaptain/ohCaptain/include/Core/ Actuator.h	499
/projects/mti/ohCaptain/ohCaptain/include/Core/ Boatswain.h	499
/projects/mti/ohCaptain/ohCaptain/include/Core/ Captain.h	500
/projects/mti/ohCaptain/ohCaptain/include/Core/ Communication.h	501
/projects/mti/ohCaptain/ohCaptain/include/Core/ constants.h	502
/projects/mti/ohCaptain/ohCaptain/include/Core/ Controller.h	504
/projects/mti/ohCaptain/ohCaptain/include/Core/ Exception.h	506
/projects/mti/ohCaptain/ohCaptain/include/Core/ literals.h	507
/projects/mti/ohCaptain/ohCaptain/include/Core/ Sensor.h	539
/projects/mti/ohCaptain/ohCaptain/include/Core/ Task.h	540
/projects/mti/ohCaptain/ohCaptain/include/Core/ Vessel.h	541
/projects/mti/ohCaptain/ohCaptain/include/Core/ World.h	542
/projects/mti/ohCaptain/ohCaptain/include/Sensors/ Gps.h	543
/projects/mti/ohCaptain/ohCaptain/include/Sensors/ KalmanIMU.h	544
/projects/mti/ohCaptain/ohCaptain/include/Sensors/ PT100.h	545
/projects/mti/ohCaptain/ohCaptain/include/Sensors/ Razor.h	546
/projects/mti/ohCaptain/ohCaptain/include/Vessels/ Meetcatamaran.h	547
/projects/mti/ohCaptain/ohCaptain/src/Communication/ LoRa_RN2483.cpp	548
/projects/mti/ohCaptain/ohCaptain/src/Controllers/ BeagleboneBlack.cpp	549
/projects/mti/ohCaptain/ohCaptain/src/Core/ Actuator.cpp	549
/projects/mti/ohCaptain/ohCaptain/src/Core/ Boatswain.cpp	549
/projects/mti/ohCaptain/ohCaptain/src/Core/ Captain.cpp	550
/projects/mti/ohCaptain/ohCaptain/src/Core/ Communication.cpp	550
/projects/mti/ohCaptain/ohCaptain/src/Core/ Controller.cpp	551
/projects/mti/ohCaptain/ohCaptain/src/Core/ Sensor.cpp	551
/projects/mti/ohCaptain/ohCaptain/src/Core/ Task.cpp	551
/projects/mti/ohCaptain/ohCaptain/src/Core/ Vessel.cpp	552
/projects/mti/ohCaptain/ohCaptain/src/Core/ World.cpp	552
/projects/mti/ohCaptain/ohCaptain/src/Sensors/ Gps.cpp	553
/projects/mti/ohCaptain/ohCaptain/src/Sensors/ KalmanIMU.cpp	553
/projects/mti/ohCaptain/ohCaptain/src/Sensors/ PT100.cpp	553
/projects/mti/ohCaptain/ohCaptain/src/Sensors/ Razor.cpp	554
/projects/mti/ohCaptain/ohCaptain/src/Vessels/ Meetcatamaran.cpp	554

Chapter 5

Namespace Documentation

5.1 boost Namespace Reference

Namespaces

- [units](#)

5.2 boost::units Namespace Reference

Namespaces

- [constants](#)
- [literals](#)

5.3 boost::units::constants Namespace Reference

Functions

- long double [operator""_LD](#) (unsigned long long x)
- long double [operator""_LD](#) (long double x)

Variables

- const auto [M_PI_LD](#) = static_cast<long double>(M_PI)
- const auto [c](#) = 299792458.0_m / 1_s
- const auto [G](#) = 6.67384E-11_m * 1_m * 1_m / 1_kg / 1_s / 1_s
- const auto [h](#) = 6.62606957E-34_J * 1_s
- const auto [hbar](#) = [h](#) / 2_LD * M_PI_LD
- const auto [u0](#) = 4E-7_N * M_PI_LD / 1_A / 1_A
- const auto [eps0](#) = 1_LD / u0 / c / c
- const auto [Z0](#) = [u0](#) * [c](#)
- const auto [ke](#) = 1_LD / (4_LD * M_PI * [eps0](#))

- const auto `e` = 1.602176565E-19_C
- const auto `me` = 9.10938291E-31_kg
- const auto `mp` = 1.672621777E-27_kg
- const auto `uB` = `e` * `hbar` / (2_LD * `me`)
- const auto `G0` = 2_LD * `e` * `e` / `h`
- const auto `KJ` = 2_LD * `e` / `h`
- const auto `uN` = `e` * `h` / (2_LD * `mp`)
- const auto `RK` = `h` / (`e` * `e`)
- const auto `alpha` = `u0` * `e` * `e` * `c` / (2_LD * `h`)
- const auto `Rinf` = `alpha` * `alpha` * `me` * `c` / (2_LD * `h`)
- const auto `a0` = `alpha` / (4_LD * M_PI_LD * `Rinf`)
- const auto `re` = `e` * `e` / (4_LD * M_PI_LD * `eps0` * `me` * `c` * `c`)
- const auto `Eh` = 2_LD * `Rinf` * `h` * `c`
- const auto `R` = 8.3144621_J / 1_K / 1_mol
- const auto `atm` = 101325_Pa
- const auto `IP` = sqrt(`hbar` * `G` / (`c` * `c` * `c`))
- const auto `mP` = sqrt(`hbar` * `c` / `G`)
- const auto `tP` = sqrt(`hbar` * `G` / (`c` * `c` * `c` * `c` * `c`))
- const auto `NA` = 6.02214129E23_LD / 1.0_mol
- const auto `k` = `R` / `NA`
- const auto `kB` = `k`
- const auto `F` = `NA` * `e`
- const auto `c1` = 2_LD * M_PI * `h` * `c` * `c`
- const auto `c2` = `h` * `c` / `k`
- const auto `sigma` = M_PI_LD * M_PI_LD * `k` * `k` * `k` * `k` / (60_LD * `hbar` * `hbar` * `hbar` * `c` * `c`)
- const auto `b` = `h` * `c` / (4.965114231_LD * `k`)
- const auto `g` = 9.80665_m / 1_s / 1_s

5.3.1 Function Documentation

5.3.1.1 operator""_LD() [1/2]

```
long double boost::units::constants::operator""_LD (
    unsigned long long x )
```

Definition at line 35 of file constants.h.

5.3.1.2 operator""_LD() [2/2]

```
long double boost::units::constants::operator""_LD (
    long double x )
```

Definition at line 36 of file constants.h.

5.3.2 Variable Documentation

5.3.2.1 a0

```
const auto boost::units::constants::a0 = alpha / (4_LD * M_PI_LD * Rinf)
```

Definition at line 57 of file constants.h.

5.3.2.2 alpha

```
const auto boost::units::constants::alpha = u0 * e * e * c / (2_LD * h)
```

Definition at line 55 of file constants.h.

5.3.2.3 atm

```
const auto boost::units::constants::atm = 101325_Pa
```

Definition at line 61 of file constants.h.

5.3.2.4 b

```
const auto boost::units::constants::b = h * c / (4.965114231_LD * k)
```

Definition at line 72 of file constants.h.

5.3.2.5 c

```
const auto boost::units::constants::c = 299792458.0_m / 1_s
```

Definition at line 39 of file constants.h.

Referenced by oCpt::LoRa::encodeTypeToHex(), oCpt::protocol::gpio::readPinValue(), and oCpt::LoRa::stringToHex().

5.3.2.6 c1

```
const auto boost::units::constants::c1 = 2_LD * M_PI * h * c * c
```

Definition at line 69 of file constants.h.

5.3.2.7 c2

```
const auto boost::units::constants::c2 = h * c / k
```

Definition at line 70 of file constants.h.

5.3.2.8 e

```
const auto boost::units::constants::e = 1.602176565E-19_C
```

Definition at line 47 of file constants.h.

Referenced by oCpt::protocol::Serial::open().

5.3.2.9 Eh

```
const auto boost::units::constants::Eh = 2_LD * Rinf * h * c
```

Definition at line 59 of file constants.h.

5.3.2.10 eps0

```
const auto boost::units::constants::eps0 = 1_LD / u0 / c / c
```

Definition at line 44 of file constants.h.

5.3.2.11 F

```
const auto boost::units::constants::F = NA * e
```

Definition at line 68 of file constants.h.

5.3.2.12 G

```
const auto boost::units::constants::G = 6.67384E-11_m * 1_m * 1_m / 1_kg / 1_s / 1_s
```

Definition at line 40 of file constants.h.

5.3.2.13 g

```
const auto boost::units::constants::g = 9.80665_m / 1_s / 1_s
```

Definition at line 73 of file constants.h.

5.3.2.14 G0

```
const auto boost::units::constants::G0 = 2_LD * e * e / h
```

Definition at line 51 of file constants.h.

5.3.2.15 h

```
const auto boost::units::constants::h = 6.62606957E-34_J * 1_s
```

Definition at line 41 of file constants.h.

5.3.2.16 hbar

```
const auto boost::units::constants::hbar = h / 2_LD * M_PI_LD
```

Definition at line 42 of file constants.h.

5.3.2.17 k

```
const auto boost::units::constants::k = R / NA
```

Definition at line 66 of file constants.h.

5.3.2.18 kB

```
const auto boost::units::constants::kB = k
```

Definition at line 67 of file constants.h.

5.3.2.19 ke

```
const auto boost::units::constants::ke = 1_LD / (4_LD * M_PI * eps0)
```

Definition at line 46 of file constants.h.

5.3.2.20 KJ

```
const auto boost::units::constants::KJ = 2_LD * e / h
```

Definition at line 52 of file constants.h.

5.3.2.21 lP

```
const auto boost::units::constants::lP = sqrt(hbar * G / (c * c * c))
```

Definition at line 62 of file constants.h.

5.3.2.22 M_PI_LD

```
const auto boost::units::constants::M_PI_LD = static_cast<long double>(M_PI)
```

Definition at line 38 of file constants.h.

5.3.2.23 me

```
const auto boost::units::constants::me = 9.10938291E-31_kg
```

Definition at line 48 of file constants.h.

5.3.2.24 mp

```
const auto boost::units::constants::mp = 1.672621777E-27_kg
```

Definition at line 49 of file constants.h.

5.3.2.25 mP

```
const auto boost::units::constants::mP = sqrt(hbar * c / G)
```

Definition at line 63 of file constants.h.

5.3.2.26 NA

```
const auto boost::units::constants::NA = 6.02214129E23_LD / 1.0_mol
```

Definition at line 65 of file constants.h.

5.3.2.27 R

```
const auto boost::units::constants::R = 8.3144621_J / 1_K / 1_mol
```

Definition at line 60 of file constants.h.

5.3.2.28 re

```
const auto boost::units::constants::re = e * e / (4_LD * M_PI_LD * eps0 * me * c * c)
```

Definition at line 58 of file constants.h.

5.3.2.29 Rinf

```
const auto boost::units::constants::Rinf = alpha * alpha * me * c / (2_LD * h)
```

Definition at line 56 of file constants.h.

5.3.2.30 RK

```
const auto boost::units::constants::RK = h / (e * e)
```

Definition at line 54 of file constants.h.

5.3.2.31 sigma

```
const auto boost::units::constants::sigma = M_PI_LD * M_PI_LD * k * k * k * k * k / (60_LD * hbar * hbar * hbar * c * c)
```

Definition at line 71 of file constants.h.

5.3.2.32 tP

```
const auto boost::units::constants::tP = sqrt(hbar * G / (c * c * c * c * c))
```

Definition at line 64 of file constants.h.

5.3.2.33 u0

```
const auto boost::units::constants::u0 = 4E-7_N * M_PI_LD / 1_A / 1_A
```

Definition at line 43 of file constants.h.

5.3.2.34 uB

```
const auto boost::units::constants::uB = e * hbar / (2_LD * me)
```

Definition at line 50 of file constants.h.

5.3.2.35 uN

```
const auto boost::units::constants::uN = e * h / (2_LD * mp)
```

Definition at line 53 of file constants.h.

5.3.2.36 Z0

```
const auto boost::units::constants::Z0 = u0 * c
```

Definition at line 45 of file constants.h.

5.4 boost::units::literals Namespace Reference

Functions

- quantity< length, long double > `operator""_Ym` (long double x)
- quantity< length, unsigned long long > `operator""_Ym` (unsigned long long x)
- quantity< length, long double > `operator""_Zm` (long double x)
- quantity< length, unsigned long long > `operator""_Zm` (unsigned long long x)
- quantity< length, long double > `operator""_Em` (long double x)
- quantity< length, unsigned long long > `operator""_Em` (unsigned long long x)
- quantity< length, long double > `operator""_Pm` (long double x)
- quantity< length, unsigned long long > `operator""_Pm` (unsigned long long x)
- quantity< length, long double > `operator""_Tm` (long double x)
- quantity< length, unsigned long long > `operator""_Tm` (unsigned long long x)
- quantity< length, long double > `operator""_Gm` (long double x)
- quantity< length, unsigned long long > `operator""_Gm` (unsigned long long x)
- quantity< length, long double > `operator""_Mm` (long double x)

- quantity< mass, long double > [operator""_ut](#) (long double x)
- quantity< mass, unsigned long long > [operator""_ut](#) (unsigned long long x)
- quantity< mass, long double > [operator""_nt](#) (long double x)
- quantity< mass, unsigned long long > [operator""_nt](#) (unsigned long long x)
- quantity< mass, long double > [operator""_pt](#) (long double x)
- quantity< mass, unsigned long long > [operator""_pt](#) (unsigned long long x)
- quantity< mass, long double > [operator""_ft](#) (long double x)
- quantity< mass, unsigned long long > [operator""_ft](#) (unsigned long long x)
- quantity< mass, long double > [operator""_at](#) (long double x)
- quantity< mass, unsigned long long > [operator""_at](#) (unsigned long long x)
- quantity< mass, long double > [operator""_zt](#) (long double x)
- quantity< mass, unsigned long long > [operator""_zt](#) (unsigned long long x)
- quantity< mass, long double > [operator""_yt](#) (long double x)
- quantity< mass, unsigned long long > [operator""_yt](#) (unsigned long long x)

5.4.1 Function Documentation

5.4.1.1 [operator""_A\(\)](#) [1/2]

```
quantity< current , long double> boost::units::literals::operator""_A ( long double x )
```

Definition at line 68 of file literals.h.

5.4.1.2 [operator""_A\(\)](#) [2/2]

```
quantity< current , unsigned long long> boost::units::literals::operator""_A ( unsigned long long x )
```

Definition at line 68 of file literals.h.

5.4.1.3 [operator""_aA\(\)](#) [1/2]

```
quantity< current , long double> boost::units::literals::operator""_aA ( long double x )
```

Definition at line 68 of file literals.h.

5.4.1.4 [operator""_aA\(\)](#) [2/2]

```
quantity< current , unsigned long long> boost::units::literals::operator""_aA ( unsigned long long x )
```

Definition at line 68 of file literals.h.

5.4.1.5 operator""_aBq() [1/2]

```
quantity< activity , long double> boost::units::literals::operator"" _aBq ( long double x )
```

Definition at line 90 of file literals.h.

5.4.1.6 operator""_aBq() [2/2]

```
quantity< activity , unsigned long long> boost::units::literals::operator"" _aBq ( unsigned long long x )
```

Definition at line 90 of file literals.h.

5.4.1.7 operator""_aC() [1/2]

```
quantity< electric_charge , long double> boost::units::literals::operator"" _aC ( long double x )
```

Definition at line 79 of file literals.h.

5.4.1.8 operator""_aC() [2/2]

```
quantity< electric_charge , unsigned long long> boost::units::literals::operator"" _aC ( unsigned long long x )
```

Definition at line 79 of file literals.h.

5.4.1.9 operator""_acd() [1/2]

```
quantity< luminous_intensity , long double> boost::units::literals::operator"" _acd ( long double x )
```

Definition at line 71 of file literals.h.

5.4.1.10 operator""_acd() [2/2]

```
quantity< luminous_intensity , unsigned long long> boost::units::literals::operator"" _acd ( unsigned long long x )
```

Definition at line 71 of file literals.h.

5.4.1.11 operator""_aday() [1/2]

```
quantity< time , long double> boost::units::literals::operator"" _aday ( long double x )
```

Definition at line 96 of file literals.h.

5.4.1.12 operator""_aday() [2/2]

```
quantity< time , unsigned long long> boost::units::literals::operator""_aday (
    unsigned long long x )
```

Definition at line 96 of file literals.h.

5.4.1.13 operator""_adeg() [1/2]

```
quantity< plane_angle , long double> boost::units::literals::operator""_adeg (
    long double x )
```

Definition at line 97 of file literals.h.

5.4.1.14 operator""_adeg() [2/2]

```
quantity< plane_angle , unsigned long long> boost::units::literals::operator""_adeg (
    unsigned long long x )
```

Definition at line 97 of file literals.h.

5.4.1.15 operator""_adegC() [1/2]

```
quantity< temperature , long double> boost::units::literals::operator""_adegC (
    long double x )
```

Definition at line 87 of file literals.h.

5.4.1.16 operator""_adegC() [2/2]

```
quantity< temperature , unsigned long long> boost::units::literals::operator""_adegC (
    unsigned long long x )
```

Definition at line 87 of file literals.h.

5.4.1.17 operator""_aF() [1/2]

```
quantity< capacitance , long double> boost::units::literals::operator""_aF (
    long double x )
```

Definition at line 81 of file literals.h.

5.4.1.18 operator""_aF() [2/2]

```
quantity< capacitance , unsigned long long> boost::units::literals::operator""_aF (
    unsigned long long x )
```

Definition at line 81 of file literals.h.

5.4.1.19 operator""_ag() [1/2]

```
quantity< mass , long double> boost::units::literals::operator"" _ag ( long double x )
```

Definition at line 66 of file literals.h.

5.4.1.20 operator""_ag() [2/2]

```
quantity< mass , unsigned long long> boost::units::literals::operator"" _ag ( unsigned long long x )
```

Definition at line 66 of file literals.h.

5.4.1.21 operator""_aGy() [1/2]

```
quantity< absorbed_dose , long double> boost::units::literals::operator"" _aGy ( long double x )
```

Definition at line 91 of file literals.h.

5.4.1.22 operator""_aGy() [2/2]

```
quantity< absorbed_dose , unsigned long long> boost::units::literals::operator"" _aGy ( unsigned long long x )
```

Definition at line 91 of file literals.h.

5.4.1.23 operator""_aH() [1/2]

```
quantity< inductance , long double> boost::units::literals::operator"" _aH ( long double x )
```

Definition at line 86 of file literals.h.

5.4.1.24 operator""_aH() [2/2]

```
quantity< inductance , unsigned long long> boost::units::literals::operator"" _aH ( unsigned long long x )
```

Definition at line 86 of file literals.h.

5.4.1.25 operator""_ah() [1/2]

```
quantity< time , unsigned long long> boost::units::literals::operator"" _ah ( unsigned long long x )
```

Definition at line 95 of file literals.h.

5.4.1.26 operator""_ah() [2/2]

```
quantity< time , long double> boost::units::literals::operator"" _ah ( long double x )
```

Definition at line 95 of file literals.h.

5.4.1.27 operator""_aHz() [1/2]

```
quantity< frequency , long double> boost::units::literals::operator"" _aHz ( long double x )
```

Definition at line 72 of file literals.h.

5.4.1.28 operator""_aHz() [2/2]

```
quantity< frequency , unsigned long long> boost::units::literals::operator"" _aHz ( unsigned long long x )
```

Definition at line 72 of file literals.h.

5.4.1.29 operator""_aJ() [1/2]

```
quantity< energy , long double> boost::units::literals::operator"" _aJ ( long double x )
```

Definition at line 77 of file literals.h.

5.4.1.30 operator""_aJ() [2/2]

```
quantity< energy , unsigned long long> boost::units::literals::operator"" _aJ ( unsigned long long x )
```

Definition at line 77 of file literals.h.

5.4.1.31 operator""_aK() [1/2]

```
quantity< temperature , unsigned long long> boost::units::literals::operator"" _aK ( unsigned long long x )
```

Definition at line 69 of file literals.h.

5.4.1.32 operator""_aK() [2/2]

```
quantity< temperature , long double> boost::units::literals::operator"" _aK ( long double x )
```

Definition at line 69 of file literals.h.

5.4.1.33 operator""_akat() [1/2]

```
quantity< catalytic_activity , long double> boost::units::literals::operator"" _akat (
    long double x )
```

Definition at line 93 of file literals.h.

5.4.1.34 operator""_akat() [2/2]

```
quantity< catalytic_activity , unsigned long long> boost::units::literals::operator"" _akat (
    unsigned long long x )
```

Definition at line 93 of file literals.h.

5.4.1.35 operator""_al() [1/2]

```
quantity< volume , long double> boost::units::literals::operator"" _al (
    long double x )
```

Definition at line 98 of file literals.h.

5.4.1.36 operator""_al() [2/2]

```
quantity< volume , unsigned long long> boost::units::literals::operator"" _al (
    unsigned long long x )
```

Definition at line 98 of file literals.h.

5.4.1.37 operator""_aL() [1/2]

```
quantity< volume , long double> boost::units::literals::operator"" _aL (
    long double x )
```

Definition at line 99 of file literals.h.

5.4.1.38 operator""_aL() [2/2]

```
quantity< volume , unsigned long long> boost::units::literals::operator"" _aL (
    unsigned long long x )
```

Definition at line 99 of file literals.h.

5.4.1.39 operator""_alm() [1/2]

```
quantity< luminous_flux , long double> boost::units::literals::operator"" _alm (
    long double x )
```

Definition at line 88 of file literals.h.

5.4.1.40 operator""_alm() [2/2]

```
quantity< luminous_flux , unsigned long long> boost::units::literals::operator""_alm ( unsigned long long x )
```

Definition at line 88 of file literals.h.

5.4.1.41 operator""_alx() [1/2]

```
quantity< illuminance , long double> boost::units::literals::operator""_alx ( long double x )
```

Definition at line 89 of file literals.h.

5.4.1.42 operator""_alx() [2/2]

```
quantity< illuminance , unsigned long long> boost::units::literals::operator""_alx ( unsigned long long x )
```

Definition at line 89 of file literals.h.

5.4.1.43 operator""_am() [1/2]

```
quantity< length , unsigned long long> boost::units::literals::operator""_am ( unsigned long long x )
```

Definition at line 65 of file literals.h.

5.4.1.44 operator""_am() [2/2]

```
quantity< length , long double> boost::units::literals::operator""_am ( long double x )
```

Definition at line 65 of file literals.h.

5.4.1.45 operator""_amin() [1/2]

```
quantity< time , long double> boost::units::literals::operator""_amin ( long double x )
```

Definition at line 94 of file literals.h.

5.4.1.46 operator""_amin() [2/2]

```
quantity< time , unsigned long long> boost::units::literals::operator""_amin ( unsigned long long x )
```

Definition at line 94 of file literals.h.

5.4.1.47 operator""_amol() [1/2]

```
quantity< amount , long double> boost::units::literals::operator"" _amol ( long double x )
```

Definition at line 70 of file literals.h.

5.4.1.48 operator""_amol() [2/2]

```
quantity< amount , unsigned long long> boost::units::literals::operator"" _amol ( unsigned long long x )
```

Definition at line 70 of file literals.h.

5.4.1.49 operator""_aN() [1/2]

```
quantity< force , long double> boost::units::literals::operator"" _aN ( long double x )
```

Definition at line 75 of file literals.h.

5.4.1.50 operator""_aN() [2/2]

```
quantity< force , unsigned long long> boost::units::literals::operator"" _aN ( unsigned long long x )
```

Definition at line 75 of file literals.h.

5.4.1.51 operator""_aohm() [1/2]

```
quantity< resistance , long double> boost::units::literals::operator"" _aohm ( long double x )
```

Definition at line 82 of file literals.h.

5.4.1.52 operator""_aohm() [2/2]

```
quantity< resistance , unsigned long long> boost::units::literals::operator"" _aohm ( unsigned long long x )
```

Definition at line 82 of file literals.h.

5.4.1.53 operator""_aPa() [1/2]

```
quantity< pressure , long double> boost::units::literals::operator"" _aPa ( long double x )
```

Definition at line 76 of file literals.h.

5.4.1.54 operator""_aPa() [2/2]

```
quantity< pressure , unsigned long long> boost::units::literals::operator"" _aPa ( unsigned long long x )
```

Definition at line 76 of file literals.h.

5.4.1.55 operator""_arad() [1/2]

```
quantity< plane_angle , long double> boost::units::literals::operator"" _arad ( long double x )
```

Definition at line 73 of file literals.h.

5.4.1.56 operator""_arad() [2/2]

```
quantity< plane_angle , unsigned long long> boost::units::literals::operator"" _arad ( unsigned long long x )
```

Definition at line 73 of file literals.h.

5.4.1.57 operator""_as() [1/2]

```
quantity< time , long double> boost::units::literals::operator"" _as ( long double x )
```

Definition at line 67 of file literals.h.

5.4.1.58 operator""_as() [2/2]

```
quantity< time , unsigned long long> boost::units::literals::operator"" _as ( unsigned long long x )
```

Definition at line 67 of file literals.h.

5.4.1.59 operator""_aS() [1/2]

```
quantity< conductance , long double> boost::units::literals::operator"" _aS ( long double x )
```

Definition at line 83 of file literals.h.

5.4.1.60 operator""_aS() [2/2]

```
quantity< conductance , unsigned long long> boost::units::literals::operator"" _aS ( unsigned long long x )
```

Definition at line 83 of file literals.h.

5.4.1.61 operator""_asr() [1/2]

```
quantity< solid_angle , long double> boost::units::literals::operator"" _asr ( long double x )
```

Definition at line 74 of file literals.h.

5.4.1.62 operator""_asr() [2/2]

```
quantity< solid_angle , unsigned long long> boost::units::literals::operator"" _asr ( unsigned long long x )
```

Definition at line 74 of file literals.h.

5.4.1.63 operator""_aSv() [1/2]

```
quantity< dose_equivalent , long double> boost::units::literals::operator"" _aSv ( long double x )
```

Definition at line 92 of file literals.h.

5.4.1.64 operator""_aSv() [2/2]

```
quantity< dose_equivalent , unsigned long long> boost::units::literals::operator"" _aSv ( unsigned long long x )
```

Definition at line 92 of file literals.h.

5.4.1.65 operator""_aT() [1/2]

```
quantity< magnetic_flux_density , long double> boost::units::literals::operator"" _aT ( long double x )
```

Definition at line 85 of file literals.h.

5.4.1.66 operator""_aT() [2/2]

```
quantity< magnetic_flux_density , unsigned long long> boost::units::literals::operator"" _aT ( unsigned long long x )
```

Definition at line 85 of file literals.h.

5.4.1.67 operator""_at() [1/2]

```
quantity< mass , long double> boost::units::literals::operator"" _at ( long double x )
```

Definition at line 100 of file literals.h.

5.4.1.68 operator""_at() [2/2]

```
quantity< mass , unsigned long long> boost::units::literals::operator"" _at (  
    unsigned long long x )
```

Definition at line 100 of file literals.h.

5.4.1.69 operator""_aV() [1/2]

```
quantity< electric_potential , long double> boost::units::literals::operator"" _aV (   
    long double x )
```

Definition at line 80 of file literals.h.

5.4.1.70 operator""_aV() [2/2]

```
quantity< electric_potential , unsigned long long> boost::units::literals::operator"" _aV (   
    unsigned long long x )
```

Definition at line 80 of file literals.h.

5.4.1.71 operator""_aW() [1/2]

```
quantity< power , long double> boost::units::literals::operator"" _aW (   
    long double x )
```

Definition at line 78 of file literals.h.

5.4.1.72 operator""_aW() [2/2]

```
quantity< power , unsigned long long> boost::units::literals::operator"" _aW (   
    unsigned long long x )
```

Definition at line 78 of file literals.h.

5.4.1.73 operator""_aWb() [1/2]

```
quantity< magnetic_flux , long double> boost::units::literals::operator"" _aWb (   
    long double x )
```

Definition at line 84 of file literals.h.

5.4.1.74 operator""_aWb() [2/2]

```
quantity< magnetic_flux , unsigned long long> boost::units::literals::operator"" _aWb (   
    unsigned long long x )
```

Definition at line 84 of file literals.h.

5.4.1.75 operator""_Bq() [1/2]

```
quantity< activity , long double> boost::units::literals::operator"" _Bq ( long double x )
```

Definition at line 90 of file literals.h.

5.4.1.76 operator""_Bq() [2/2]

```
quantity< activity , unsigned long long> boost::units::literals::operator"" _Bq ( unsigned long long x )
```

Definition at line 90 of file literals.h.

5.4.1.77 operator""_C() [1/2]

```
quantity< electric_charge , unsigned long long> boost::units::literals::operator"" _C ( unsigned long long x )
```

Definition at line 79 of file literals.h.

5.4.1.78 operator""_C() [2/2]

```
quantity< electric_charge , long double> boost::units::literals::operator"" _C ( long double x )
```

Definition at line 79 of file literals.h.

5.4.1.79 operator""_cA() [1/2]

```
quantity< current , long double> boost::units::literals::operator"" _cA ( long double x )
```

Definition at line 68 of file literals.h.

5.4.1.80 operator""_cA() [2/2]

```
quantity< current , unsigned long long> boost::units::literals::operator"" _cA ( unsigned long long x )
```

Definition at line 68 of file literals.h.

5.4.1.81 operator""_cBq() [1/2]

```
quantity< activity , long double> boost::units::literals::operator"" _cBq ( long double x )
```

Definition at line 90 of file literals.h.

5.4.1.82 operator""_cBq() [2/2]

```
quantity< activity , unsigned long long> boost::units::literals::operator"" _cBq (  
    unsigned long long x )
```

Definition at line 90 of file literals.h.

5.4.1.83 operator""_cC() [1/2]

```
quantity< electric_charge , long double> boost::units::literals::operator"" _cC (   
    long double x )
```

Definition at line 79 of file literals.h.

5.4.1.84 operator""_cC() [2/2]

```
quantity< electric_charge , unsigned long long> boost::units::literals::operator"" _cC (   
    unsigned long long x )
```

Definition at line 79 of file literals.h.

5.4.1.85 operator""_ccd() [1/2]

```
quantity< luminous_intensity , long double> boost::units::literals::operator"" _ccd (   
    long double x )
```

Definition at line 71 of file literals.h.

5.4.1.86 operator""_ccd() [2/2]

```
quantity< luminous_intensity , unsigned long long> boost::units::literals::operator"" _ccd (   
    unsigned long long x )
```

Definition at line 71 of file literals.h.

5.4.1.87 operator""_cd() [1/2]

```
quantity< luminous_intensity , long double> boost::units::literals::operator"" _cd (   
    long double x )
```

Definition at line 71 of file literals.h.

5.4.1.88 operator""_cd() [2/2]

```
quantity< luminous_intensity , unsigned long long> boost::units::literals::operator"" _cd (   
    unsigned long long x )
```

Definition at line 71 of file literals.h.

5.4.1.89 operator""_cday() [1/2]

```
quantity< time , long double> boost::units::literals::operator"" _cday ( long double x )
```

Definition at line 96 of file literals.h.

5.4.1.90 operator""_cday() [2/2]

```
quantity< time , unsigned long long> boost::units::literals::operator"" _cday ( unsigned long long x )
```

Definition at line 96 of file literals.h.

5.4.1.91 operator""_cdeg() [1/2]

```
quantity< plane_angle , unsigned long long> boost::units::literals::operator"" _cdeg ( unsigned long long x )
```

Definition at line 97 of file literals.h.

5.4.1.92 operator""_cdeg() [2/2]

```
quantity< plane_angle , long double> boost::units::literals::operator"" _cdeg ( long double x )
```

Definition at line 97 of file literals.h.

5.4.1.93 operator""_cdegC() [1/2]

```
quantity< temperature , long double> boost::units::literals::operator"" _cdegC ( long double x )
```

Definition at line 87 of file literals.h.

5.4.1.94 operator""_cdegC() [2/2]

```
quantity< temperature , unsigned long long> boost::units::literals::operator"" _cdegC ( unsigned long long x )
```

Definition at line 87 of file literals.h.

5.4.1.95 operator""_cF() [1/2]

```
quantity< capacitance , long double> boost::units::literals::operator"" _cF ( long double x )
```

Definition at line 81 of file literals.h.

5.4.1.96 operator""_cF() [2/2]

```
quantity< capacitance , unsigned long long> boost::units::literals::operator"" _cF ( unsigned long long x )
```

Definition at line 81 of file literals.h.

5.4.1.97 operator""_cg() [1/2]

```
quantity< mass , unsigned long long> boost::units::literals::operator"" _cg ( unsigned long long x )
```

Definition at line 66 of file literals.h.

5.4.1.98 operator""_cg() [2/2]

```
quantity< mass , long double> boost::units::literals::operator"" _cg ( long double x )
```

Definition at line 66 of file literals.h.

5.4.1.99 operator""_cGy() [1/2]

```
quantity< absorbed_dose , long double> boost::units::literals::operator"" _cGy ( long double x )
```

Definition at line 91 of file literals.h.

5.4.1.100 operator""_cGy() [2/2]

```
quantity< absorbed_dose , unsigned long long> boost::units::literals::operator"" _cGy ( unsigned long long x )
```

Definition at line 91 of file literals.h.

5.4.1.101 operator""_cH() [1/2]

```
quantity< inductance , long double> boost::units::literals::operator"" _cH ( long double x )
```

Definition at line 86 of file literals.h.

5.4.1.102 operator""_cH() [2/2]

```
quantity< inductance , unsigned long long> boost::units::literals::operator"" _cH ( unsigned long long x )
```

Definition at line 86 of file literals.h.

5.4.1.103 operator""_ch() [1/2]

```
quantity< time , unsigned long long> boost::units::literals::operator"" _ch (  
    unsigned long long x )
```

Definition at line 95 of file literals.h.

5.4.1.104 operator""_ch() [2/2]

```
quantity< time , long double> boost::units::literals::operator"" _ch (   
    long double x )
```

Definition at line 95 of file literals.h.

5.4.1.105 operator""_cHz() [1/2]

```
quantity< frequency , long double> boost::units::literals::operator"" _cHz (   
    long double x )
```

Definition at line 72 of file literals.h.

5.4.1.106 operator""_cHz() [2/2]

```
quantity< frequency , unsigned long long> boost::units::literals::operator"" _cHz (   
    unsigned long long x )
```

Definition at line 72 of file literals.h.

5.4.1.107 operator""_cJ() [1/2]

```
quantity< energy , long double> boost::units::literals::operator"" _cJ (   
    long double x )
```

Definition at line 77 of file literals.h.

5.4.1.108 operator""_cJ() [2/2]

```
quantity< energy , unsigned long long> boost::units::literals::operator"" _cJ (   
    unsigned long long x )
```

Definition at line 77 of file literals.h.

5.4.1.109 operator""_cK() [1/2]

```
quantity< temperature , long double> boost::units::literals::operator"" _cK (   
    long double x )
```

Definition at line 69 of file literals.h.

5.4.1.110 operator""_cK() [2/2]

```
quantity< temperature , unsigned long long> boost::units::literals::operator"" _cK ( unsigned long long x )
```

Definition at line 69 of file literals.h.

5.4.1.111 operator""_ckat() [1/2]

```
quantity< catalytic_activity , long double> boost::units::literals::operator"" _ckat ( long double x )
```

Definition at line 93 of file literals.h.

5.4.1.112 operator""_ckat() [2/2]

```
quantity< catalytic_activity , unsigned long long> boost::units::literals::operator"" _ckat ( unsigned long long x )
```

Definition at line 93 of file literals.h.

5.4.1.113 operator""_cl() [1/2]

```
quantity< volume , unsigned long long> boost::units::literals::operator"" _cl ( unsigned long long x )
```

Definition at line 98 of file literals.h.

5.4.1.114 operator""_cl() [2/2]

```
quantity< volume , long double> boost::units::literals::operator"" _cl ( long double x )
```

Definition at line 98 of file literals.h.

5.4.1.115 operator""_cL() [1/2]

```
quantity< volume , unsigned long long> boost::units::literals::operator"" _cL ( unsigned long long x )
```

Definition at line 99 of file literals.h.

5.4.1.116 operator""_cL() [2/2]

```
quantity< volume , long double> boost::units::literals::operator"" _cL ( long double x )
```

Definition at line 99 of file literals.h.

5.4.1.117 operator""_clm() [1/2]

```
quantity< luminous_flux , long double> boost::units::literals::operator"" _clm (
    long double x )
```

Definition at line 88 of file literals.h.

5.4.1.118 operator""_clm() [2/2]

```
quantity< luminous_flux , unsigned long long> boost::units::literals::operator"" _clm (
    unsigned long long x )
```

Definition at line 88 of file literals.h.

5.4.1.119 operator""_clk() [1/2]

```
quantity< illuminance , unsigned long long> boost::units::literals::operator"" _clk (
    unsigned long long x )
```

Definition at line 89 of file literals.h.

5.4.1.120 operator""_clk() [2/2]

```
quantity< illuminance , long double> boost::units::literals::operator"" _clk (
    long double x )
```

Definition at line 89 of file literals.h.

5.4.1.121 operator""_cm() [1/2]

```
quantity< length , long double> boost::units::literals::operator"" _cm (
    long double x )
```

Definition at line 65 of file literals.h.

5.4.1.122 operator""_cm() [2/2]

```
quantity< length , unsigned long long> boost::units::literals::operator"" _cm (
    unsigned long long x )
```

Definition at line 65 of file literals.h.

5.4.1.123 operator""_cmin() [1/2]

```
quantity< time , long double> boost::units::literals::operator"" _cmin (
    long double x )
```

Definition at line 94 of file literals.h.

5.4.1.124 operator"" _cmin() [2/2]

```
quantity< time , unsigned long long> boost::units::literals::operator"" _cmin (  
    unsigned long long x )
```

Definition at line 94 of file literals.h.

5.4.1.125 operator"" _cmol() [1/2]

```
quantity< amount , long double> boost::units::literals::operator"" _cmol (   
    long double x )
```

Definition at line 70 of file literals.h.

5.4.1.126 operator"" _cmol() [2/2]

```
quantity< amount , unsigned long long> boost::units::literals::operator"" _cmol (   
    unsigned long long x )
```

Definition at line 70 of file literals.h.

5.4.1.127 operator"" _cN() [1/2]

```
quantity< force , long double> boost::units::literals::operator"" _cN (   
    long double x )
```

Definition at line 75 of file literals.h.

5.4.1.128 operator"" _cN() [2/2]

```
quantity< force , unsigned long long> boost::units::literals::operator"" _cN (   
    unsigned long long x )
```

Definition at line 75 of file literals.h.

5.4.1.129 operator"" _coh() [1/2]

```
quantity< resistance , unsigned long long> boost::units::literals::operator"" _coh (   
    unsigned long long x )
```

Definition at line 82 of file literals.h.

5.4.1.130 operator"" _coh() [2/2]

```
quantity< resistance , long double> boost::units::literals::operator"" _coh (   
    long double x )
```

Definition at line 82 of file literals.h.

5.4.1.131 operator""_cPa() [1/2]

```
quantity< pressure , long double> boost::units::literals::operator"" _cPa ( long double x )
```

Definition at line 76 of file literals.h.

5.4.1.132 operator""_cPa() [2/2]

```
quantity< pressure , unsigned long long> boost::units::literals::operator"" _cPa ( unsigned long long x )
```

Definition at line 76 of file literals.h.

5.4.1.133 operator""_crad() [1/2]

```
quantity< plane_angle , unsigned long long> boost::units::literals::operator"" _crad ( unsigned long long x )
```

Definition at line 73 of file literals.h.

5.4.1.134 operator""_crad() [2/2]

```
quantity< plane_angle , long double> boost::units::literals::operator"" _crad ( long double x )
```

Definition at line 73 of file literals.h.

5.4.1.135 operator""_cs() [1/2]

```
quantity< time , long double> boost::units::literals::operator"" _cs ( long double x )
```

Definition at line 67 of file literals.h.

5.4.1.136 operator""_cs() [2/2]

```
quantity< time , unsigned long long> boost::units::literals::operator"" _cs ( unsigned long long x )
```

Definition at line 67 of file literals.h.

5.4.1.137 operator""_cS() [1/2]

```
quantity< conductance , long double> boost::units::literals::operator"" _cS ( long double x )
```

Definition at line 83 of file literals.h.

5.4.1.138 operator""_cS() [2/2]

```
quantity< conductance , unsigned long long> boost::units::literals::operator"" _cS ( unsigned long long x )
```

Definition at line 83 of file literals.h.

5.4.1.139 operator""_csr() [1/2]

```
quantity< solid_angle , long double> boost::units::literals::operator"" _csr ( long double x )
```

Definition at line 74 of file literals.h.

5.4.1.140 operator""_csr() [2/2]

```
quantity< solid_angle , unsigned long long> boost::units::literals::operator"" _csr ( unsigned long long x )
```

Definition at line 74 of file literals.h.

5.4.1.141 operator""_cSv() [1/2]

```
quantity< dose_equivalent , long double> boost::units::literals::operator"" _cSv ( long double x )
```

Definition at line 92 of file literals.h.

5.4.1.142 operator""_cSv() [2/2]

```
quantity< dose_equivalent , unsigned long long> boost::units::literals::operator"" _cSv ( unsigned long long x )
```

Definition at line 92 of file literals.h.

5.4.1.143 operator""_cT() [1/2]

```
quantity< magnetic_flux_density , unsigned long long> boost::units::literals::operator"" _cT ( unsigned long long x )
```

Definition at line 85 of file literals.h.

5.4.1.144 operator""_cT() [2/2]

```
quantity< magnetic_flux_density , long double> boost::units::literals::operator"" _cT ( long double x )
```

Definition at line 85 of file literals.h.

5.4.1.145 operator""_ct() [1/2]

```
quantity< mass , long double> boost::units::literals::operator"" _ct ( long double x )
```

Definition at line 100 of file literals.h.

5.4.1.146 operator""_ct() [2/2]

```
quantity< mass , unsigned long long> boost::units::literals::operator"" _ct ( unsigned long long x )
```

Definition at line 100 of file literals.h.

5.4.1.147 operator""_cV() [1/2]

```
quantity< electric_potential , long double> boost::units::literals::operator"" _cV ( long double x )
```

Definition at line 80 of file literals.h.

5.4.1.148 operator""_cV() [2/2]

```
quantity< electric_potential , unsigned long long> boost::units::literals::operator"" _cV ( unsigned long long x )
```

Definition at line 80 of file literals.h.

5.4.1.149 operator""_cW() [1/2]

```
quantity< power , long double> boost::units::literals::operator"" _cW ( long double x )
```

Definition at line 78 of file literals.h.

5.4.1.150 operator""_cW() [2/2]

```
quantity< power , unsigned long long> boost::units::literals::operator"" _cW ( unsigned long long x )
```

Definition at line 78 of file literals.h.

5.4.1.151 operator""_cWb() [1/2]

```
quantity< magnetic_flux , long double> boost::units::literals::operator"" _cWb ( long double x )
```

Definition at line 84 of file literals.h.

5.4.1.152 operator"" _cWb() [2/2]

```
quantity< magnetic_flux , unsigned long long> boost::units::literals::operator"" _cWb ( unsigned long long x )
```

Definition at line 84 of file literals.h.

5.4.1.153 operator"" _dA() [1/2]

```
quantity< current , long double> boost::units::literals::operator"" _dA ( long double x )
```

Definition at line 68 of file literals.h.

5.4.1.154 operator"" _dA() [2/2]

```
quantity< current , unsigned long long> boost::units::literals::operator"" _dA ( unsigned long long x )
```

Definition at line 68 of file literals.h.

5.4.1.155 operator"" _daA() [1/2]

```
quantity< current , long double> boost::units::literals::operator"" _daA ( long double x )
```

Definition at line 68 of file literals.h.

5.4.1.156 operator"" _daA() [2/2]

```
quantity< current , unsigned long long> boost::units::literals::operator"" _daA ( unsigned long long x )
```

Definition at line 68 of file literals.h.

5.4.1.157 operator"" _daBq() [1/2]

```
quantity< activity , long double> boost::units::literals::operator"" _daBq ( long double x )
```

Definition at line 90 of file literals.h.

5.4.1.158 operator"" _daBq() [2/2]

```
quantity< activity , unsigned long long> boost::units::literals::operator"" _daBq ( unsigned long long x )
```

Definition at line 90 of file literals.h.

5.4.1.159 operator""_daC() [1/2]

```
quantity< electric_charge , long double> boost::units::literals::operator"" _daC ( long double x )
```

Definition at line 79 of file literals.h.

5.4.1.160 operator""_daC() [2/2]

```
quantity< electric_charge , unsigned long long> boost::units::literals::operator"" _daC ( unsigned long long x )
```

Definition at line 79 of file literals.h.

5.4.1.161 operator""_dacd() [1/2]

```
quantity< luminous_intensity , long double> boost::units::literals::operator"" _dacd ( long double x )
```

Definition at line 71 of file literals.h.

5.4.1.162 operator""_dacd() [2/2]

```
quantity< luminous_intensity , unsigned long long> boost::units::literals::operator"" _dacd ( unsigned long long x )
```

Definition at line 71 of file literals.h.

5.4.1.163 operator""_daday() [1/2]

```
quantity< time , unsigned long long> boost::units::literals::operator"" _daday ( unsigned long long x )
```

Definition at line 96 of file literals.h.

5.4.1.164 operator""_daday() [2/2]

```
quantity< time , long double> boost::units::literals::operator"" _daday ( long double x )
```

Definition at line 96 of file literals.h.

5.4.1.165 operator""_dadeg() [1/2]

```
quantity< plane_angle , long double> boost::units::literals::operator"" _dadeg ( long double x )
```

Definition at line 97 of file literals.h.

5.4.1.166 operator""_dadeG() [2/2]

```
quantity< plane_angle , unsigned long long> boost::units::literals::operator"" _dadeG ( unsigned long long x )
```

Definition at line 97 of file literals.h.

5.4.1.167 operator""_dadeGc() [1/2]

```
quantity< temperature , long double> boost::units::literals::operator"" _dadeGc ( long double x )
```

Definition at line 87 of file literals.h.

5.4.1.168 operator""_dadeGc() [2/2]

```
quantity< temperature , unsigned long long> boost::units::literals::operator"" _dadeGc ( unsigned long long x )
```

Definition at line 87 of file literals.h.

5.4.1.169 operator""_daF() [1/2]

```
quantity< capacitance , long double> boost::units::literals::operator"" _daF ( long double x )
```

Definition at line 81 of file literals.h.

5.4.1.170 operator""_daF() [2/2]

```
quantity< capacitance , unsigned long long> boost::units::literals::operator"" _daF ( unsigned long long x )
```

Definition at line 81 of file literals.h.

5.4.1.171 operator""_dag() [1/2]

```
quantity< mass , long double> boost::units::literals::operator"" _dag ( long double x )
```

Definition at line 66 of file literals.h.

5.4.1.172 operator""_dag() [2/2]

```
quantity< mass , unsigned long long> boost::units::literals::operator"" _dag ( unsigned long long x )
```

Definition at line 66 of file literals.h.

5.4.1.173 operator""_daGy() [1/2]

```
quantity< absorbed_dose , long double> boost::units::literals::operator"" _daGy ( long double x )
```

Definition at line 91 of file literals.h.

5.4.1.174 operator""_daGy() [2/2]

```
quantity< absorbed_dose , unsigned long long> boost::units::literals::operator"" _daGy ( unsigned long long x )
```

Definition at line 91 of file literals.h.

5.4.1.175 operator""_daH() [1/2]

```
quantity< inductance , unsigned long long> boost::units::literals::operator"" _daH ( unsigned long long x )
```

Definition at line 86 of file literals.h.

5.4.1.176 operator""_daH() [2/2]

```
quantity< inductance , long double> boost::units::literals::operator"" _daH ( long double x )
```

Definition at line 86 of file literals.h.

5.4.1.177 operator""_dah() [1/2]

```
quantity< time , long double> boost::units::literals::operator"" _dah ( long double x )
```

Definition at line 95 of file literals.h.

5.4.1.178 operator""_dah() [2/2]

```
quantity< time , unsigned long long> boost::units::literals::operator"" _dah ( unsigned long long x )
```

Definition at line 95 of file literals.h.

5.4.1.179 operator""_daHz() [1/2]

```
quantity< frequency , unsigned long long> boost::units::literals::operator"" _daHz ( unsigned long long x )
```

Definition at line 72 of file literals.h.

5.4.1.180 operator"" _daHz() [2/2]

```
quantity< frequency , long double> boost::units::literals::operator"" _daHz ( long double x )
```

Definition at line 72 of file literals.h.

5.4.1.181 operator"" _daJ() [1/2]

```
quantity< energy , long double> boost::units::literals::operator"" _daJ ( long double x )
```

Definition at line 77 of file literals.h.

5.4.1.182 operator"" _daJ() [2/2]

```
quantity< energy , unsigned long long> boost::units::literals::operator"" _daJ ( unsigned long long x )
```

Definition at line 77 of file literals.h.

5.4.1.183 operator"" _daK() [1/2]

```
quantity< temperature , long double> boost::units::literals::operator"" _daK ( long double x )
```

Definition at line 69 of file literals.h.

5.4.1.184 operator"" _daK() [2/2]

```
quantity< temperature , unsigned long long> boost::units::literals::operator"" _daK ( unsigned long long x )
```

Definition at line 69 of file literals.h.

5.4.1.185 operator"" _dakat() [1/2]

```
quantity< catalytic_activity , long double> boost::units::literals::operator"" _dakat ( long double x )
```

Definition at line 93 of file literals.h.

5.4.1.186 operator"" _dakat() [2/2]

```
quantity< catalytic_activity , unsigned long long> boost::units::literals::operator"" _dakat ( unsigned long long x )
```

Definition at line 93 of file literals.h.

5.4.1.187 operator""_dal() [1/2]

```
quantity< volume , long double> boost::units::literals::operator"" _dal ( long double x )
```

Definition at line 98 of file literals.h.

5.4.1.188 operator""_dal() [2/2]

```
quantity< volume , unsigned long long> boost::units::literals::operator"" _dal ( unsigned long long x )
```

Definition at line 98 of file literals.h.

5.4.1.189 operator""_daL() [1/2]

```
quantity< volume , long double> boost::units::literals::operator"" _daL ( long double x )
```

Definition at line 99 of file literals.h.

5.4.1.190 operator""_daL() [2/2]

```
quantity< volume , unsigned long long> boost::units::literals::operator"" _daL ( unsigned long long x )
```

Definition at line 99 of file literals.h.

5.4.1.191 operator""_dalm() [1/2]

```
quantity< luminous_flux , long double> boost::units::literals::operator"" _dalm ( long double x )
```

Definition at line 88 of file literals.h.

5.4.1.192 operator""_dalm() [2/2]

```
quantity< luminous_flux , unsigned long long> boost::units::literals::operator"" _dalm ( unsigned long long x )
```

Definition at line 88 of file literals.h.

5.4.1.193 operator""_dalx() [1/2]

```
quantity< illuminance , unsigned long long> boost::units::literals::operator"" _dalx ( unsigned long long x )
```

Definition at line 89 of file literals.h.

5.4.1.194 operator"" _dalx() [2/2]

```
quantity< illuminance , long double> boost::units::literals::operator"" _dalx ( long double x )
```

Definition at line 89 of file literals.h.

5.4.1.195 operator"" _dam() [1/2]

```
quantity< length , unsigned long long> boost::units::literals::operator"" _dam ( unsigned long long x )
```

Definition at line 65 of file literals.h.

5.4.1.196 operator"" _dam() [2/2]

```
quantity< length , long double> boost::units::literals::operator"" _dam ( long double x )
```

Definition at line 65 of file literals.h.

5.4.1.197 operator"" _damin() [1/2]

```
quantity< time , unsigned long long> boost::units::literals::operator"" _damin ( unsigned long long x )
```

Definition at line 94 of file literals.h.

5.4.1.198 operator"" _damin() [2/2]

```
quantity< time , long double> boost::units::literals::operator"" _damin ( long double x )
```

Definition at line 94 of file literals.h.

5.4.1.199 operator"" _damol() [1/2]

```
quantity< amount , unsigned long long> boost::units::literals::operator"" _damol ( unsigned long long x )
```

Definition at line 70 of file literals.h.

5.4.1.200 operator"" _damol() [2/2]

```
quantity< amount , long double> boost::units::literals::operator"" _damol ( long double x )
```

Definition at line 70 of file literals.h.

5.4.1.201 operator""_daN() [1/2]

```
quantity< force , long double> boost::units::literals::operator"" _daN ( long double x )
```

Definition at line 75 of file literals.h.

5.4.1.202 operator""_daN() [2/2]

```
quantity< force , unsigned long long> boost::units::literals::operator"" _daN ( unsigned long long x )
```

Definition at line 75 of file literals.h.

5.4.1.203 operator""_daohm() [1/2]

```
quantity< resistance , long double> boost::units::literals::operator"" _daohm ( long double x )
```

Definition at line 82 of file literals.h.

5.4.1.204 operator""_daohm() [2/2]

```
quantity< resistance , unsigned long long> boost::units::literals::operator"" _daohm ( unsigned long long x )
```

Definition at line 82 of file literals.h.

5.4.1.205 operator""_daPa() [1/2]

```
quantity< pressure , unsigned long long> boost::units::literals::operator"" _daPa ( unsigned long long x )
```

Definition at line 76 of file literals.h.

5.4.1.206 operator""_daPa() [2/2]

```
quantity< pressure , long double> boost::units::literals::operator"" _daPa ( long double x )
```

Definition at line 76 of file literals.h.

5.4.1.207 operator""_darad() [1/2]

```
quantity< plane_angle , unsigned long long> boost::units::literals::operator"" _darad ( unsigned long long x )
```

Definition at line 73 of file literals.h.

5.4.1.208 operator""_darad() [2/2]

```
quantity< plane_angle , long double> boost::units::literals::operator""_darad ( long double x )
```

Definition at line 73 of file literals.h.

5.4.1.209 operator""_das() [1/2]

```
quantity< time , long double> boost::units::literals::operator""_das ( long double x )
```

Definition at line 67 of file literals.h.

5.4.1.210 operator""_das() [2/2]

```
quantity< time , unsigned long long> boost::units::literals::operator""_das ( unsigned long long x )
```

Definition at line 67 of file literals.h.

5.4.1.211 operator""_daS() [1/2]

```
quantity< conductance , long double> boost::units::literals::operator""_daS ( long double x )
```

Definition at line 83 of file literals.h.

5.4.1.212 operator""_daS() [2/2]

```
quantity< conductance , unsigned long long> boost::units::literals::operator""_daS ( unsigned long long x )
```

Definition at line 83 of file literals.h.

5.4.1.213 operator""_dasr() [1/2]

```
quantity< solid_angle , long double> boost::units::literals::operator""_dasr ( long double x )
```

Definition at line 74 of file literals.h.

5.4.1.214 operator""_dasr() [2/2]

```
quantity< solid_angle , unsigned long long> boost::units::literals::operator""_dasr ( unsigned long long x )
```

Definition at line 74 of file literals.h.

5.4.1.215 operator""_daSv() [1/2]

```
quantity< dose_equivalent , unsigned long long> boost::units::literals::operator"" _daSv ( unsigned long long x )
```

Definition at line 92 of file literals.h.

5.4.1.216 operator""_daSv() [2/2]

```
quantity< dose_equivalent , long double> boost::units::literals::operator"" _daSv ( long double x )
```

Definition at line 92 of file literals.h.

5.4.1.217 operator""_daT() [1/2]

```
quantity< magnetic_flux_density , long double> boost::units::literals::operator"" _daT ( long double x )
```

Definition at line 85 of file literals.h.

5.4.1.218 operator""_daT() [2/2]

```
quantity< magnetic_flux_density , unsigned long long> boost::units::literals::operator"" _daT ( unsigned long long x )
```

Definition at line 85 of file literals.h.

5.4.1.219 operator""_dat() [1/2]

```
quantity< mass , unsigned long long> boost::units::literals::operator"" _dat ( unsigned long long x )
```

Definition at line 100 of file literals.h.

5.4.1.220 operator""_dat() [2/2]

```
quantity< mass , long double> boost::units::literals::operator"" _dat ( long double x )
```

Definition at line 100 of file literals.h.

5.4.1.221 operator""_daV() [1/2]

```
quantity< electric_potential , unsigned long long> boost::units::literals::operator"" _daV ( unsigned long long x )
```

Definition at line 80 of file literals.h.

5.4.1.222 operator"" _daV() [2/2]

```
quantity< electric_potential , long double> boost::units::literals::operator"" _daV ( long double x )
```

Definition at line 80 of file literals.h.

5.4.1.223 operator"" _daW() [1/2]

```
quantity< power , unsigned long long> boost::units::literals::operator"" _daW ( unsigned long long x )
```

Definition at line 78 of file literals.h.

5.4.1.224 operator"" _daW() [2/2]

```
quantity< power , long double> boost::units::literals::operator"" _daW ( long double x )
```

Definition at line 78 of file literals.h.

5.4.1.225 operator"" _daWb() [1/2]

```
quantity< magnetic_flux , unsigned long long> boost::units::literals::operator"" _daWb ( unsigned long long x )
```

Definition at line 84 of file literals.h.

5.4.1.226 operator"" _daWb() [2/2]

```
quantity< magnetic_flux , long double> boost::units::literals::operator"" _daWb ( long double x )
```

Definition at line 84 of file literals.h.

5.4.1.227 operator"" _day() [1/2]

```
quantity< time , long double> boost::units::literals::operator"" _day ( long double x )
```

Definition at line 96 of file literals.h.

5.4.1.228 operator"" _day() [2/2]

```
quantity< time , unsigned long long> boost::units::literals::operator"" _day ( unsigned long long x )
```

Definition at line 96 of file literals.h.

5.4.1.229 operator""_dBq() [1/2]

```
quantity< activity , long double> boost::units::literals::operator"" _dBq ( long double x )
```

Definition at line 90 of file literals.h.

5.4.1.230 operator""_dBq() [2/2]

```
quantity< activity , unsigned long long> boost::units::literals::operator"" _dBq ( unsigned long long x )
```

Definition at line 90 of file literals.h.

5.4.1.231 operator""_dC() [1/2]

```
quantity< electric_charge , long double> boost::units::literals::operator"" _dC ( long double x )
```

Definition at line 79 of file literals.h.

5.4.1.232 operator""_dC() [2/2]

```
quantity< electric_charge , unsigned long long> boost::units::literals::operator"" _dC ( unsigned long long x )
```

Definition at line 79 of file literals.h.

5.4.1.233 operator""_dcd() [1/2]

```
quantity< luminous_intensity , long double> boost::units::literals::operator"" _dcd ( long double x )
```

Definition at line 71 of file literals.h.

5.4.1.234 operator""_dcd() [2/2]

```
quantity< luminous_intensity , unsigned long long> boost::units::literals::operator"" _dcd ( unsigned long long x )
```

Definition at line 71 of file literals.h.

5.4.1.235 operator""_dday() [1/2]

```
quantity< time , long double> boost::units::literals::operator"" _dday ( long double x )
```

Definition at line 96 of file literals.h.

5.4.1.236 operator""_dday() [2/2]

```
quantity< time , unsigned long long> boost::units::literals::operator"" _dday (  
    unsigned long long x )
```

Definition at line 96 of file literals.h.

5.4.1.237 operator""_ddeg() [1/2]

```
quantity< plane_angle , long double> boost::units::literals::operator"" _ddeg (   
    long double x )
```

Definition at line 97 of file literals.h.

5.4.1.238 operator""_ddeg() [2/2]

```
quantity< plane_angle , unsigned long long> boost::units::literals::operator"" _ddeg (   
    unsigned long long x )
```

Definition at line 97 of file literals.h.

5.4.1.239 operator""_ddegC() [1/2]

```
quantity< temperature , long double> boost::units::literals::operator"" _ddegC (   
    long double x )
```

Definition at line 87 of file literals.h.

5.4.1.240 operator""_ddegC() [2/2]

```
quantity< temperature , unsigned long long> boost::units::literals::operator"" _ddegC (   
    unsigned long long x )
```

Definition at line 87 of file literals.h.

5.4.1.241 operator""_deg() [1/2]

```
quantity< plane_angle , long double> boost::units::literals::operator"" _deg (   
    long double x )
```

Definition at line 97 of file literals.h.

5.4.1.242 operator""_deg() [2/2]

```
quantity< plane_angle , unsigned long long> boost::units::literals::operator"" _deg (   
    unsigned long long x )
```

Definition at line 97 of file literals.h.

5.4.1.243 operator""_degC() [1/2]

```
quantity< temperature , long double> boost::units::literals::operator"" _degC ( long double x )
```

Definition at line 87 of file literals.h.

5.4.1.244 operator""_degC() [2/2]

```
quantity< temperature , unsigned long long> boost::units::literals::operator"" _degC ( unsigned long long x )
```

Definition at line 87 of file literals.h.

5.4.1.245 operator""_dF() [1/2]

```
quantity< capacitance , long double> boost::units::literals::operator"" _dF ( long double x )
```

Definition at line 81 of file literals.h.

5.4.1.246 operator""_dF() [2/2]

```
quantity< capacitance , unsigned long long> boost::units::literals::operator"" _dF ( unsigned long long x )
```

Definition at line 81 of file literals.h.

5.4.1.247 operator""_dg() [1/2]

```
quantity< mass , long double> boost::units::literals::operator"" _dg ( long double x )
```

Definition at line 66 of file literals.h.

5.4.1.248 operator""_dg() [2/2]

```
quantity< mass , unsigned long long> boost::units::literals::operator"" _dg ( unsigned long long x )
```

Definition at line 66 of file literals.h.

5.4.1.249 operator""_dGy() [1/2]

```
quantity< absorbed_dose , long double> boost::units::literals::operator"" _dGy ( long double x )
```

Definition at line 91 of file literals.h.

5.4.1.250 operator"" _dGy() [2/2]

```
quantity< absorbed_dose , unsigned long long> boost::units::literals::operator"" _dGy ( unsigned long long x )
```

Definition at line 91 of file literals.h.

5.4.1.251 operator"" _dH() [1/2]

```
quantity< inductance , long double> boost::units::literals::operator"" _dH ( long double x )
```

Definition at line 86 of file literals.h.

5.4.1.252 operator"" _dH() [2/2]

```
quantity< inductance , unsigned long long> boost::units::literals::operator"" _dH ( unsigned long long x )
```

Definition at line 86 of file literals.h.

5.4.1.253 operator"" _dh() [1/2]

```
quantity< time , unsigned long long> boost::units::literals::operator"" _dh ( unsigned long long x )
```

Definition at line 95 of file literals.h.

5.4.1.254 operator"" _dh() [2/2]

```
quantity< time , long double> boost::units::literals::operator"" _dh ( long double x )
```

Definition at line 95 of file literals.h.

5.4.1.255 operator"" _dHz() [1/2]

```
quantity< frequency , long double> boost::units::literals::operator"" _dHz ( long double x )
```

Definition at line 72 of file literals.h.

5.4.1.256 operator"" _dHz() [2/2]

```
quantity< frequency , unsigned long long> boost::units::literals::operator"" _dHz ( unsigned long long x )
```

Definition at line 72 of file literals.h.

5.4.1.257 operator""_dJ() [1/2]

```
quantity< energy , long double> boost::units::literals::operator"" _dJ ( long double x )
```

Definition at line 77 of file literals.h.

5.4.1.258 operator""_dJ() [2/2]

```
quantity< energy , unsigned long long> boost::units::literals::operator"" _dJ ( unsigned long long x )
```

Definition at line 77 of file literals.h.

5.4.1.259 operator""_dK() [1/2]

```
quantity< temperature , long double> boost::units::literals::operator"" _dK ( long double x )
```

Definition at line 69 of file literals.h.

5.4.1.260 operator""_dK() [2/2]

```
quantity< temperature , unsigned long long> boost::units::literals::operator"" _dK ( unsigned long long x )
```

Definition at line 69 of file literals.h.

5.4.1.261 operator""_dkat() [1/2]

```
quantity< catalytic_activity , long double> boost::units::literals::operator"" _dkat ( long double x )
```

Definition at line 93 of file literals.h.

5.4.1.262 operator""_dkat() [2/2]

```
quantity< catalytic_activity , unsigned long long> boost::units::literals::operator"" _dkat ( unsigned long long x )
```

Definition at line 93 of file literals.h.

5.4.1.263 operator""_dl() [1/2]

```
quantity< volume , long double> boost::units::literals::operator"" _dl ( long double x )
```

Definition at line 98 of file literals.h.

5.4.1.264 operator""_dl() [2/2]

```
quantity< volume , unsigned long long> boost::units::literals::operator"" _dl (  
    unsigned long long x )
```

Definition at line 98 of file literals.h.

5.4.1.265 operator""_dL() [1/2]

```
quantity< volume , long double> boost::units::literals::operator"" _dL (   
    long double x )
```

Definition at line 99 of file literals.h.

5.4.1.266 operator""_dL() [2/2]

```
quantity< volume , unsigned long long> boost::units::literals::operator"" _dL (   
    unsigned long long x )
```

Definition at line 99 of file literals.h.

5.4.1.267 operator""_dlm() [1/2]

```
quantity< luminous_flux , long double> boost::units::literals::operator"" _dlm (   
    long double x )
```

Definition at line 88 of file literals.h.

5.4.1.268 operator""_dlm() [2/2]

```
quantity< luminous_flux , unsigned long long> boost::units::literals::operator"" _dlm (   
    unsigned long long x )
```

Definition at line 88 of file literals.h.

5.4.1.269 operator""_dlx() [1/2]

```
quantity< illuminance , long double> boost::units::literals::operator"" _dlx (   
    long double x )
```

Definition at line 89 of file literals.h.

5.4.1.270 operator""_dlx() [2/2]

```
quantity< illuminance , unsigned long long> boost::units::literals::operator"" _dlx (   
    unsigned long long x )
```

Definition at line 89 of file literals.h.

5.4.1.271 operator""_dm() [1/2]

```
quantity< length , long double> boost::units::literals::operator"" _dm ( long double x )
```

Definition at line 65 of file literals.h.

5.4.1.272 operator""_dm() [2/2]

```
quantity< length , unsigned long long> boost::units::literals::operator"" _dm ( unsigned long long x )
```

Definition at line 65 of file literals.h.

5.4.1.273 operator""_dmin() [1/2]

```
quantity< time , long double> boost::units::literals::operator"" _dmin ( long double x )
```

Definition at line 94 of file literals.h.

5.4.1.274 operator""_dmin() [2/2]

```
quantity< time , unsigned long long> boost::units::literals::operator"" _dmin ( unsigned long long x )
```

Definition at line 94 of file literals.h.

5.4.1.275 operator""_dmol() [1/2]

```
quantity< amount , long double> boost::units::literals::operator"" _dmol ( long double x )
```

Definition at line 70 of file literals.h.

5.4.1.276 operator""_dmol() [2/2]

```
quantity< amount , unsigned long long> boost::units::literals::operator"" _dmol ( unsigned long long x )
```

Definition at line 70 of file literals.h.

5.4.1.277 operator""_dN() [1/2]

```
quantity< force , long double> boost::units::literals::operator"" _dN ( long double x )
```

Definition at line 75 of file literals.h.

5.4.1.278 operator"" _dN() [2/2]

```
quantity< force , unsigned long long> boost::units::literals::operator"" _dN (  
    unsigned long long x )
```

Definition at line 75 of file literals.h.

5.4.1.279 operator"" _dohm() [1/2]

```
quantity< resistance , long double> boost::units::literals::operator"" _dohm (   
    long double x )
```

Definition at line 82 of file literals.h.

5.4.1.280 operator"" _dohm() [2/2]

```
quantity< resistance , unsigned long long> boost::units::literals::operator"" _dohm (   
    unsigned long long x )
```

Definition at line 82 of file literals.h.

5.4.1.281 operator"" _dPa() [1/2]

```
quantity< pressure , long double> boost::units::literals::operator"" _dPa (   
    long double x )
```

Definition at line 76 of file literals.h.

5.4.1.282 operator"" _dPa() [2/2]

```
quantity< pressure , unsigned long long> boost::units::literals::operator"" _dPa (   
    unsigned long long x )
```

Definition at line 76 of file literals.h.

5.4.1.283 operator"" _drad() [1/2]

```
quantity< plane_angle , long double> boost::units::literals::operator"" _drad (   
    long double x )
```

Definition at line 73 of file literals.h.

5.4.1.284 operator"" _drad() [2/2]

```
quantity< plane_angle , unsigned long long> boost::units::literals::operator"" _drad (   
    unsigned long long x )
```

Definition at line 73 of file literals.h.

5.4.1.285 operator""_ds() [1/2]

```
quantity< time , long double> boost::units::literals::operator"" _ds ( long double x )
```

Definition at line 67 of file literals.h.

5.4.1.286 operator""_ds() [2/2]

```
quantity< time , unsigned long long> boost::units::literals::operator"" _ds ( unsigned long long x )
```

Definition at line 67 of file literals.h.

5.4.1.287 operator""_dS() [1/2]

```
quantity< conductance , long double> boost::units::literals::operator"" _dS ( long double x )
```

Definition at line 83 of file literals.h.

5.4.1.288 operator""_dS() [2/2]

```
quantity< conductance , unsigned long long> boost::units::literals::operator"" _dS ( unsigned long long x )
```

Definition at line 83 of file literals.h.

5.4.1.289 operator""_dsr() [1/2]

```
quantity< solid_angle , long double> boost::units::literals::operator"" _dsr ( long double x )
```

Definition at line 74 of file literals.h.

5.4.1.290 operator""_dsr() [2/2]

```
quantity< solid_angle , unsigned long long> boost::units::literals::operator"" _dsr ( unsigned long long x )
```

Definition at line 74 of file literals.h.

5.4.1.291 operator""_dSv() [1/2]

```
quantity< dose_equivalent , long double> boost::units::literals::operator"" _dSv ( long double x )
```

Definition at line 92 of file literals.h.

5.4.1.292 operator""_dSv() [2/2]

```
quantity< dose_equivalent , unsigned long long> boost::units::literals::operator"" _dSv ( unsigned long long x )
```

Definition at line 92 of file literals.h.

5.4.1.293 operator""_dT() [1/2]

```
quantity< magnetic_flux_density , long double> boost::units::literals::operator"" _dT ( long double x )
```

Definition at line 85 of file literals.h.

5.4.1.294 operator""_dT() [2/2]

```
quantity< magnetic_flux_density , unsigned long long> boost::units::literals::operator"" _dT ( unsigned long long x )
```

Definition at line 85 of file literals.h.

5.4.1.295 operator""_dt() [1/2]

```
quantity< mass , unsigned long long> boost::units::literals::operator"" _dt ( unsigned long long x )
```

Definition at line 100 of file literals.h.

5.4.1.296 operator""_dt() [2/2]

```
quantity< mass , long double> boost::units::literals::operator"" _dt ( long double x )
```

Definition at line 100 of file literals.h.

5.4.1.297 operator""_dV() [1/2]

```
quantity< electric_potential , long double> boost::units::literals::operator"" _dV ( long double x )
```

Definition at line 80 of file literals.h.

5.4.1.298 operator""_dV() [2/2]

```
quantity< electric_potential , unsigned long long> boost::units::literals::operator"" _dV ( unsigned long long x )
```

Definition at line 80 of file literals.h.

5.4.1.299 operator""_dW() [1/2]

```
quantity< power , long double> boost::units::literals::operator"" _dW ( long double x )
```

Definition at line 78 of file literals.h.

5.4.1.300 operator""_dW() [2/2]

```
quantity< power , unsigned long long> boost::units::literals::operator"" _dW ( unsigned long long x )
```

Definition at line 78 of file literals.h.

5.4.1.301 operator""_dWb() [1/2]

```
quantity< magnetic_flux , long double> boost::units::literals::operator"" _dWb ( long double x )
```

Definition at line 84 of file literals.h.

5.4.1.302 operator""_dWb() [2/2]

```
quantity< magnetic_flux , unsigned long long> boost::units::literals::operator"" _dWb ( unsigned long long x )
```

Definition at line 84 of file literals.h.

5.4.1.303 operator""_EA() [1/2]

```
quantity< current , long double> boost::units::literals::operator"" _EA ( long double x )
```

Definition at line 68 of file literals.h.

5.4.1.304 operator""_EA() [2/2]

```
quantity< current , unsigned long long> boost::units::literals::operator"" _EA ( unsigned long long x )
```

Definition at line 68 of file literals.h.

5.4.1.305 operator""_EBq() [1/2]

```
quantity< activity , long double> boost::units::literals::operator"" _EBq ( long double x )
```

Definition at line 90 of file literals.h.

5.4.1.306 operator"" _EBq() [2/2]

```
quantity< activity , unsigned long long> boost::units::literals::operator"" _EBq ( unsigned long long x )
```

Definition at line 90 of file literals.h.

5.4.1.307 operator"" _EC() [1/2]

```
quantity< electric_charge , long double> boost::units::literals::operator"" _EC ( long double x )
```

Definition at line 79 of file literals.h.

5.4.1.308 operator"" _EC() [2/2]

```
quantity< electric_charge , unsigned long long> boost::units::literals::operator"" _EC ( unsigned long long x )
```

Definition at line 79 of file literals.h.

5.4.1.309 operator"" _Ecd() [1/2]

```
quantity< luminous_intensity , long double> boost::units::literals::operator"" _Ecd ( long double x )
```

Definition at line 71 of file literals.h.

5.4.1.310 operator"" _Ecd() [2/2]

```
quantity< luminous_intensity , unsigned long long> boost::units::literals::operator"" _Ecd ( unsigned long long x )
```

Definition at line 71 of file literals.h.

5.4.1.311 operator"" _Eday() [1/2]

```
quantity< time , long double> boost::units::literals::operator"" _Eday ( long double x )
```

Definition at line 96 of file literals.h.

5.4.1.312 operator"" _Eday() [2/2]

```
quantity< time , unsigned long long> boost::units::literals::operator"" _Eday ( unsigned long long x )
```

Definition at line 96 of file literals.h.

5.4.1.313 operator""_Edeg() [1/2]

```
quantity< plane_angle , unsigned long long> boost::units::literals::operator""_Edeg (  
    unsigned long long x )
```

Definition at line 97 of file literals.h.

5.4.1.314 operator""_Edeg() [2/2]

```
quantity< plane_angle , long double> boost::units::literals::operator""_Edeg (   
    long double x )
```

Definition at line 97 of file literals.h.

5.4.1.315 operator""_EdegC() [1/2]

```
quantity< temperature , long double> boost::units::literals::operator""_EdegC (   
    long double x )
```

Definition at line 87 of file literals.h.

5.4.1.316 operator""_EdegC() [2/2]

```
quantity< temperature , unsigned long long> boost::units::literals::operator""_EdegC (   
    unsigned long long x )
```

Definition at line 87 of file literals.h.

5.4.1.317 operator""_EF() [1/2]

```
quantity< capacitance , unsigned long long> boost::units::literals::operator""_EF (   
    unsigned long long x )
```

Definition at line 81 of file literals.h.

5.4.1.318 operator""_EF() [2/2]

```
quantity< capacitance , long double> boost::units::literals::operator""_EF (   
    long double x )
```

Definition at line 81 of file literals.h.

5.4.1.319 operator""_Eg() [1/2]

```
quantity< mass , long double> boost::units::literals::operator""_Eg (   
    long double x )
```

Definition at line 66 of file literals.h.

5.4.1.320 operator"" _Eg() [2/2]

```
quantity< mass , unsigned long long> boost::units::literals::operator"" _Eg (  
    unsigned long long x )
```

Definition at line 66 of file literals.h.

5.4.1.321 operator"" _EGy() [1/2]

```
quantity< absorbed_dose , long double> boost::units::literals::operator"" _EGy (   
    long double x )
```

Definition at line 91 of file literals.h.

5.4.1.322 operator"" _EGy() [2/2]

```
quantity< absorbed_dose , unsigned long long> boost::units::literals::operator"" _EGy (   
    unsigned long long x )
```

Definition at line 91 of file literals.h.

5.4.1.323 operator"" _EH() [1/2]

```
quantity< inductance , long double> boost::units::literals::operator"" _EH (   
    long double x )
```

Definition at line 86 of file literals.h.

5.4.1.324 operator"" _EH() [2/2]

```
quantity< inductance , unsigned long long> boost::units::literals::operator"" _EH (   
    unsigned long long x )
```

Definition at line 86 of file literals.h.

5.4.1.325 operator"" _Eh() [1/2]

```
quantity< time , long double> boost::units::literals::operator"" _Eh (   
    long double x )
```

Definition at line 95 of file literals.h.

5.4.1.326 operator"" _Eh() [2/2]

```
quantity< time , unsigned long long> boost::units::literals::operator"" _Eh (   
    unsigned long long x )
```

Definition at line 95 of file literals.h.

5.4.1.327 operator""_EHz() [1/2]

```
quantity< frequency , long double> boost::units::literals::operator"" _EHz ( long double x )
```

Definition at line 72 of file literals.h.

5.4.1.328 operator""_EHz() [2/2]

```
quantity< frequency , unsigned long long> boost::units::literals::operator"" _EHz ( unsigned long long x )
```

Definition at line 72 of file literals.h.

5.4.1.329 operator""_EJ() [1/2]

```
quantity< energy , long double> boost::units::literals::operator"" _EJ ( long double x )
```

Definition at line 77 of file literals.h.

5.4.1.330 operator""_EJ() [2/2]

```
quantity< energy , unsigned long long> boost::units::literals::operator"" _EJ ( unsigned long long x )
```

Definition at line 77 of file literals.h.

5.4.1.331 operator""_EK() [1/2]

```
quantity< temperature , long double> boost::units::literals::operator"" _EK ( long double x )
```

Definition at line 69 of file literals.h.

5.4.1.332 operator""_EK() [2/2]

```
quantity< temperature , unsigned long long> boost::units::literals::operator"" _EK ( unsigned long long x )
```

Definition at line 69 of file literals.h.

5.4.1.333 operator""_Ekat() [1/2]

```
quantity< catalytic_activity , long double> boost::units::literals::operator"" _Ekat ( long double x )
```

Definition at line 93 of file literals.h.

5.4.1.334 operator"" _Ekat() [2/2]

```
quantity< catalytic_activity , unsigned long long> boost::units::literals::operator"" _Ekat (  
    unsigned long long x )
```

Definition at line 93 of file literals.h.

5.4.1.335 operator"" _El() [1/2]

```
quantity< volume , long double> boost::units::literals::operator"" _El (   
    long double x )
```

Definition at line 98 of file literals.h.

5.4.1.336 operator"" _El() [2/2]

```
quantity< volume , unsigned long long> boost::units::literals::operator"" _El (   
    unsigned long long x )
```

Definition at line 98 of file literals.h.

5.4.1.337 operator"" _EL() [1/2]

```
quantity< volume , long double> boost::units::literals::operator"" _EL (   
    long double x )
```

Definition at line 99 of file literals.h.

5.4.1.338 operator"" _EL() [2/2]

```
quantity< volume , unsigned long long> boost::units::literals::operator"" _EL (   
    unsigned long long x )
```

Definition at line 99 of file literals.h.

5.4.1.339 operator"" _Elm() [1/2]

```
quantity< luminous_flux , long double> boost::units::literals::operator"" _Elm (   
    long double x )
```

Definition at line 88 of file literals.h.

5.4.1.340 operator"" _Elm() [2/2]

```
quantity< luminous_flux , unsigned long long> boost::units::literals::operator"" _Elm (   
    unsigned long long x )
```

Definition at line 88 of file literals.h.

5.4.1.341 operator""_Elx() [1/2]

```
quantity< illuminance , long double> boost::units::literals::operator""_Elx ( long double x )
```

Definition at line 89 of file literals.h.

5.4.1.342 operator""_Elx() [2/2]

```
quantity< illuminance , unsigned long long> boost::units::literals::operator""_Elx ( unsigned long long x )
```

Definition at line 89 of file literals.h.

5.4.1.343 operator""_Em() [1/2]

```
quantity< length , long double> boost::units::literals::operator""_Em ( long double x )
```

Definition at line 65 of file literals.h.

5.4.1.344 operator""_Em() [2/2]

```
quantity< length , unsigned long long> boost::units::literals::operator""_Em ( unsigned long long x )
```

Definition at line 65 of file literals.h.

5.4.1.345 operator""_Emin() [1/2]

```
quantity< time , long double> boost::units::literals::operator""_Emin ( long double x )
```

Definition at line 94 of file literals.h.

5.4.1.346 operator""_Emin() [2/2]

```
quantity< time , unsigned long long> boost::units::literals::operator""_Emin ( unsigned long long x )
```

Definition at line 94 of file literals.h.

5.4.1.347 operator""_Emol() [1/2]

```
quantity< amount , long double> boost::units::literals::operator""_Emol ( long double x )
```

Definition at line 70 of file literals.h.

5.4.1.348 operator"" _Emol() [2/2]

```
quantity< amount , unsigned long long> boost::units::literals::operator"" _Emol (  
    unsigned long long x )
```

Definition at line 70 of file literals.h.

5.4.1.349 operator"" _EN() [1/2]

```
quantity< force , long double> boost::units::literals::operator"" _EN (   
    long double x )
```

Definition at line 75 of file literals.h.

5.4.1.350 operator"" _EN() [2/2]

```
quantity< force , unsigned long long> boost::units::literals::operator"" _EN (   
    unsigned long long x )
```

Definition at line 75 of file literals.h.

5.4.1.351 operator"" _Eohm() [1/2]

```
quantity< resistance , long double> boost::units::literals::operator"" _Eohm (   
    long double x )
```

Definition at line 82 of file literals.h.

5.4.1.352 operator"" _Eohm() [2/2]

```
quantity< resistance , unsigned long long> boost::units::literals::operator"" _Eohm (   
    unsigned long long x )
```

Definition at line 82 of file literals.h.

5.4.1.353 operator"" _EPa() [1/2]

```
quantity< pressure , long double> boost::units::literals::operator"" _EPa (   
    long double x )
```

Definition at line 76 of file literals.h.

5.4.1.354 operator"" _EPa() [2/2]

```
quantity< pressure , unsigned long long> boost::units::literals::operator"" _EPa (   
    unsigned long long x )
```

Definition at line 76 of file literals.h.

5.4.1.355 operator""_Erad() [1/2]

```
quantity< plane_angle , long double> boost::units::literals::operator""_Erad ( long double x )
```

Definition at line 73 of file literals.h.

5.4.1.356 operator""_Erad() [2/2]

```
quantity< plane_angle , unsigned long long> boost::units::literals::operator""_Erad ( unsigned long long x )
```

Definition at line 73 of file literals.h.

5.4.1.357 operator""_Es() [1/2]

```
quantity< time , long double> boost::units::literals::operator""_Es ( long double x )
```

Definition at line 67 of file literals.h.

5.4.1.358 operator""_Es() [2/2]

```
quantity< time , unsigned long long> boost::units::literals::operator""_Es ( unsigned long long x )
```

Definition at line 67 of file literals.h.

5.4.1.359 operator""_ES() [1/2]

```
quantity< conductance , long double> boost::units::literals::operator""_ES ( long double x )
```

Definition at line 83 of file literals.h.

5.4.1.360 operator""_ES() [2/2]

```
quantity< conductance , unsigned long long> boost::units::literals::operator""_ES ( unsigned long long x )
```

Definition at line 83 of file literals.h.

5.4.1.361 operator""_Esr() [1/2]

```
quantity< solid_angle , long double> boost::units::literals::operator""_Esr ( long double x )
```

Definition at line 74 of file literals.h.

5.4.1.362 operator"" _Esr() [2/2]

```
quantity< solid_angle , unsigned long long> boost::units::literals::operator"" _Esr (  
    unsigned long long x )
```

Definition at line 74 of file literals.h.

5.4.1.363 operator"" _ESv() [1/2]

```
quantity< dose_equivalent , long double> boost::units::literals::operator"" _ESv (   
    long double x )
```

Definition at line 92 of file literals.h.

5.4.1.364 operator"" _ESv() [2/2]

```
quantity< dose_equivalent , unsigned long long> boost::units::literals::operator"" _ESv (   
    unsigned long long x )
```

Definition at line 92 of file literals.h.

5.4.1.365 operator"" _ET() [1/2]

```
quantity< magnetic_flux_density , long double> boost::units::literals::operator"" _ET (   
    long double x )
```

Definition at line 85 of file literals.h.

5.4.1.366 operator"" _ET() [2/2]

```
quantity< magnetic_flux_density , unsigned long long> boost::units::literals::operator"" _ET (   
    unsigned long long x )
```

Definition at line 85 of file literals.h.

5.4.1.367 operator"" _Et() [1/2]

```
quantity< mass , long double> boost::units::literals::operator"" _Et (   
    long double x )
```

Definition at line 100 of file literals.h.

5.4.1.368 operator"" _Et() [2/2]

```
quantity< mass , unsigned long long> boost::units::literals::operator"" _Et (   
    unsigned long long x )
```

Definition at line 100 of file literals.h.

5.4.1.369 operator""_EV() [1/2]

```
quantity< electric_potential , long double> boost::units::literals::operator""_EV ( long double x )
```

Definition at line 80 of file literals.h.

5.4.1.370 operator""_EV() [2/2]

```
quantity< electric_potential , unsigned long long> boost::units::literals::operator""_EV ( unsigned long long x )
```

Definition at line 80 of file literals.h.

5.4.1.371 operator""_EW() [1/2]

```
quantity< power , long double> boost::units::literals::operator""_EW ( long double x )
```

Definition at line 78 of file literals.h.

5.4.1.372 operator""_EW() [2/2]

```
quantity< power , unsigned long long> boost::units::literals::operator""_EW ( unsigned long long x )
```

Definition at line 78 of file literals.h.

5.4.1.373 operator""_EWb() [1/2]

```
quantity< magnetic_flux , long double> boost::units::literals::operator""_EWb ( long double x )
```

Definition at line 84 of file literals.h.

5.4.1.374 operator""_EWb() [2/2]

```
quantity< magnetic_flux , unsigned long long> boost::units::literals::operator""_EWb ( unsigned long long x )
```

Definition at line 84 of file literals.h.

5.4.1.375 operator""_F() [1/2]

```
quantity< capacitance , unsigned long long> boost::units::literals::operator""_F ( unsigned long long x )
```

Definition at line 81 of file literals.h.

5.4.1.376 operator""_F() [2/2]

```
quantity< capacitance , long double> boost::units::literals::operator""_F ( long double x )
```

Definition at line 81 of file literals.h.

5.4.1.377 operator""_fA() [1/2]

```
quantity< current , long double> boost::units::literals::operator""_fA ( long double x )
```

Definition at line 68 of file literals.h.

5.4.1.378 operator""_fA() [2/2]

```
quantity< current , unsigned long long> boost::units::literals::operator""_fA ( unsigned long long x )
```

Definition at line 68 of file literals.h.

5.4.1.379 operator""_fBq() [1/2]

```
quantity< activity , long double> boost::units::literals::operator""_fBq ( long double x )
```

Definition at line 90 of file literals.h.

5.4.1.380 operator""_fBq() [2/2]

```
quantity< activity , unsigned long long> boost::units::literals::operator""_fBq ( unsigned long long x )
```

Definition at line 90 of file literals.h.

5.4.1.381 operator""_fC() [1/2]

```
quantity< electric_charge , long double> boost::units::literals::operator""_fC ( long double x )
```

Definition at line 79 of file literals.h.

5.4.1.382 operator""_fC() [2/2]

```
quantity< electric_charge , unsigned long long> boost::units::literals::operator""_fC ( unsigned long long x )
```

Definition at line 79 of file literals.h.

5.4.1.383 operator""_fcd() [1/2]

```
quantity< luminous_intensity , long double> boost::units::literals::operator"" _fcd (
    long double x )
```

Definition at line 71 of file literals.h.

5.4.1.384 operator""_fcd() [2/2]

```
quantity< luminous_intensity , unsigned long long> boost::units::literals::operator"" _fcd (
    unsigned long long x )
```

Definition at line 71 of file literals.h.

5.4.1.385 operator""_fday() [1/2]

```
quantity< time , long double> boost::units::literals::operator"" _fday (
    long double x )
```

Definition at line 96 of file literals.h.

5.4.1.386 operator""_fday() [2/2]

```
quantity< time , unsigned long long> boost::units::literals::operator"" _fday (
    unsigned long long x )
```

Definition at line 96 of file literals.h.

5.4.1.387 operator""_fdeg() [1/2]

```
quantity< plane_angle , unsigned long long> boost::units::literals::operator"" _fdeg (
    unsigned long long x )
```

Definition at line 97 of file literals.h.

5.4.1.388 operator""_fdeg() [2/2]

```
quantity< plane_angle , long double> boost::units::literals::operator"" _fdeg (
    long double x )
```

Definition at line 97 of file literals.h.

5.4.1.389 operator""_fdegC() [1/2]

```
quantity< temperature , long double> boost::units::literals::operator"" _fdegC (
    long double x )
```

Definition at line 87 of file literals.h.

5.4.1.390 operator"" _fdegC() [2/2]

```
quantity< temperature , unsigned long long> boost::units::literals::operator"" _fdegC (  
    unsigned long long x )
```

Definition at line 87 of file literals.h.

5.4.1.391 operator"" _fF() [1/2]

```
quantity< capacitance , unsigned long long> boost::units::literals::operator"" _fF (   
    unsigned long long x )
```

Definition at line 81 of file literals.h.

5.4.1.392 operator"" _fF() [2/2]

```
quantity< capacitance , long double> boost::units::literals::operator"" _fF (   
    long double x )
```

Definition at line 81 of file literals.h.

5.4.1.393 operator"" _fg() [1/2]

```
quantity< mass , long double> boost::units::literals::operator"" _fg (   
    long double x )
```

Definition at line 66 of file literals.h.

5.4.1.394 operator"" _fg() [2/2]

```
quantity< mass , unsigned long long> boost::units::literals::operator"" _fg (   
    unsigned long long x )
```

Definition at line 66 of file literals.h.

5.4.1.395 operator"" _fGy() [1/2]

```
quantity< absorbed_dose , long double> boost::units::literals::operator"" _fGy (   
    long double x )
```

Definition at line 91 of file literals.h.

5.4.1.396 operator"" _fGy() [2/2]

```
quantity< absorbed_dose , unsigned long long> boost::units::literals::operator"" _fGy (   
    unsigned long long x )
```

Definition at line 91 of file literals.h.

5.4.1.397 operator""_fH() [1/2]

```
quantity< inductance , long double> boost::units::literals::operator"" _fH ( long double x )
```

Definition at line 86 of file literals.h.

5.4.1.398 operator""_fH() [2/2]

```
quantity< inductance , unsigned long long> boost::units::literals::operator"" _fH ( unsigned long long x )
```

Definition at line 86 of file literals.h.

5.4.1.399 operator""_fh() [1/2]

```
quantity< time , long double> boost::units::literals::operator"" _fh ( long double x )
```

Definition at line 95 of file literals.h.

5.4.1.400 operator""_fh() [2/2]

```
quantity< time , unsigned long long> boost::units::literals::operator"" _fh ( unsigned long long x )
```

Definition at line 95 of file literals.h.

5.4.1.401 operator""_fHz() [1/2]

```
quantity< frequency , long double> boost::units::literals::operator"" _fHz ( long double x )
```

Definition at line 72 of file literals.h.

5.4.1.402 operator""_fHz() [2/2]

```
quantity< frequency , unsigned long long> boost::units::literals::operator"" _fHz ( unsigned long long x )
```

Definition at line 72 of file literals.h.

5.4.1.403 operator""_fJ() [1/2]

```
quantity< energy , long double> boost::units::literals::operator"" _fJ ( long double x )
```

Definition at line 77 of file literals.h.

5.4.1.404 operator""_fJ() [2/2]

```
quantity< energy , unsigned long long> boost::units::literals::operator"" _fJ (  
    unsigned long long x )
```

Definition at line 77 of file literals.h.

5.4.1.405 operator""_fK() [1/2]

```
quantity< temperature , long double> boost::units::literals::operator"" _fK (   
    long double x )
```

Definition at line 69 of file literals.h.

5.4.1.406 operator""_fK() [2/2]

```
quantity< temperature , unsigned long long> boost::units::literals::operator"" _fK (   
    unsigned long long x )
```

Definition at line 69 of file literals.h.

5.4.1.407 operator""_fkat() [1/2]

```
quantity< catalytic_activity , long double> boost::units::literals::operator"" _fkat (   
    long double x )
```

Definition at line 93 of file literals.h.

5.4.1.408 operator""_fkat() [2/2]

```
quantity< catalytic_activity , unsigned long long> boost::units::literals::operator"" _fkat (   
    unsigned long long x )
```

Definition at line 93 of file literals.h.

5.4.1.409 operator""_fl() [1/2]

```
quantity< volume , long double> boost::units::literals::operator"" _fl (   
    long double x )
```

Definition at line 98 of file literals.h.

5.4.1.410 operator""_fl() [2/2]

```
quantity< volume , unsigned long long> boost::units::literals::operator"" _fl (   
    unsigned long long x )
```

Definition at line 98 of file literals.h.

5.4.1.411 operator""_fL() [1/2]

```
quantity< volume , long double> boost::units::literals::operator"" _fL ( long double x )
```

Definition at line 99 of file literals.h.

5.4.1.412 operator""_fL() [2/2]

```
quantity< volume , unsigned long long> boost::units::literals::operator"" _fL ( unsigned long long x )
```

Definition at line 99 of file literals.h.

5.4.1.413 operator""_flm() [1/2]

```
quantity< luminous_flux , long double> boost::units::literals::operator"" _flm ( long double x )
```

Definition at line 88 of file literals.h.

5.4.1.414 operator""_flm() [2/2]

```
quantity< luminous_flux , unsigned long long> boost::units::literals::operator"" _flm ( unsigned long long x )
```

Definition at line 88 of file literals.h.

5.4.1.415 operator""_flx() [1/2]

```
quantity< illuminance , unsigned long long> boost::units::literals::operator"" _flx ( unsigned long long x )
```

Definition at line 89 of file literals.h.

5.4.1.416 operator""_flx() [2/2]

```
quantity< illuminance , long double> boost::units::literals::operator"" _flx ( long double x )
```

Definition at line 89 of file literals.h.

5.4.1.417 operator""_fm() [1/2]

```
quantity< length , unsigned long long> boost::units::literals::operator"" _fm ( unsigned long long x )
```

Definition at line 65 of file literals.h.

5.4.1.418 operator""_fm() [2/2]

```
quantity< length , long double> boost::units::literals::operator"" _fm ( long double x )
```

Definition at line 65 of file literals.h.

5.4.1.419 operator""_fmin() [1/2]

```
quantity< time , long double> boost::units::literals::operator"" _fmin ( long double x )
```

Definition at line 94 of file literals.h.

5.4.1.420 operator""_fmin() [2/2]

```
quantity< time , unsigned long long> boost::units::literals::operator"" _fmin ( unsigned long long x )
```

Definition at line 94 of file literals.h.

5.4.1.421 operator""_fmol() [1/2]

```
quantity< amount , long double> boost::units::literals::operator"" _fmol ( long double x )
```

Definition at line 70 of file literals.h.

5.4.1.422 operator""_fmol() [2/2]

```
quantity< amount , unsigned long long> boost::units::literals::operator"" _fmol ( unsigned long long x )
```

Definition at line 70 of file literals.h.

5.4.1.423 operator""_fN() [1/2]

```
quantity< force , long double> boost::units::literals::operator"" _fN ( long double x )
```

Definition at line 75 of file literals.h.

5.4.1.424 operator""_fN() [2/2]

```
quantity< force , unsigned long long> boost::units::literals::operator"" _fN ( unsigned long long x )
```

Definition at line 75 of file literals.h.

5.4.1.425 operator""_fohm() [1/2]

```
quantity< resistance , long double> boost::units::literals::operator"" _fohm ( long double x )
```

Definition at line 82 of file literals.h.

5.4.1.426 operator""_fohm() [2/2]

```
quantity< resistance , unsigned long long> boost::units::literals::operator"" _fohm ( unsigned long long x )
```

Definition at line 82 of file literals.h.

5.4.1.427 operator""_fPa() [1/2]

```
quantity< pressure , long double> boost::units::literals::operator"" _fPa ( long double x )
```

Definition at line 76 of file literals.h.

5.4.1.428 operator""_fPa() [2/2]

```
quantity< pressure , unsigned long long> boost::units::literals::operator"" _fPa ( unsigned long long x )
```

Definition at line 76 of file literals.h.

5.4.1.429 operator""_frad() [1/2]

```
quantity< plane_angle , long double> boost::units::literals::operator"" _frad ( long double x )
```

Definition at line 73 of file literals.h.

5.4.1.430 operator""_frad() [2/2]

```
quantity< plane_angle , unsigned long long> boost::units::literals::operator"" _frad ( unsigned long long x )
```

Definition at line 73 of file literals.h.

5.4.1.431 operator""_fs() [1/2]

```
quantity< time , long double> boost::units::literals::operator"" _fs ( long double x )
```

Definition at line 67 of file literals.h.

5.4.1.432 operator""_fs() [2/2]

```
quantity< time , unsigned long long> boost::units::literals::operator""_fs (  
    unsigned long long x )
```

Definition at line 67 of file literals.h.

5.4.1.433 operator""_fS() [1/2]

```
quantity< conductance , long double> boost::units::literals::operator""_fS (   
    long double x )
```

Definition at line 83 of file literals.h.

5.4.1.434 operator""_fS() [2/2]

```
quantity< conductance , unsigned long long> boost::units::literals::operator""_fS (   
    unsigned long long x )
```

Definition at line 83 of file literals.h.

5.4.1.435 operator""_fsr() [1/2]

```
quantity< solid_angle , long double> boost::units::literals::operator""_fsr (   
    long double x )
```

Definition at line 74 of file literals.h.

5.4.1.436 operator""_fsr() [2/2]

```
quantity< solid_angle , unsigned long long> boost::units::literals::operator""_fsr (   
    unsigned long long x )
```

Definition at line 74 of file literals.h.

5.4.1.437 operator""_fSv() [1/2]

```
quantity< dose_equivalent , long double> boost::units::literals::operator""_fSv (   
    long double x )
```

Definition at line 92 of file literals.h.

5.4.1.438 operator""_fSv() [2/2]

```
quantity< dose_equivalent , unsigned long long> boost::units::literals::operator""_fSv (   
    unsigned long long x )
```

Definition at line 92 of file literals.h.

5.4.1.439 operator""_fT() [1/2]

```
quantity< magnetic_flux_density , long double> boost::units::literals::operator"" _fT ( long double x )
```

Definition at line 85 of file literals.h.

5.4.1.440 operator""_fT() [2/2]

```
quantity< magnetic_flux_density , unsigned long long> boost::units::literals::operator"" _fT ( unsigned long long x )
```

Definition at line 85 of file literals.h.

5.4.1.441 operator""_ft() [1/2]

```
quantity< mass , long double> boost::units::literals::operator"" _ft ( long double x )
```

Definition at line 100 of file literals.h.

5.4.1.442 operator""_ft() [2/2]

```
quantity< mass , unsigned long long> boost::units::literals::operator"" _ft ( unsigned long long x )
```

Definition at line 100 of file literals.h.

5.4.1.443 operator""_fV() [1/2]

```
quantity< electric_potential , long double> boost::units::literals::operator"" _fV ( long double x )
```

Definition at line 80 of file literals.h.

5.4.1.444 operator""_fV() [2/2]

```
quantity< electric_potential , unsigned long long> boost::units::literals::operator"" _fV ( unsigned long long x )
```

Definition at line 80 of file literals.h.

5.4.1.445 operator""_fW() [1/2]

```
quantity< power , long double> boost::units::literals::operator"" _fW ( long double x )
```

Definition at line 78 of file literals.h.

5.4.1.446 operator""_fW() [2/2]

```
quantity< power , unsigned long long> boost::units::literals::operator"" _fW (  
    unsigned long long x )
```

Definition at line 78 of file literals.h.

5.4.1.447 operator""_fWb() [1/2]

```
quantity< magnetic_flux , long double> boost::units::literals::operator"" _fWb (   
    long double x )
```

Definition at line 84 of file literals.h.

5.4.1.448 operator""_fWb() [2/2]

```
quantity< magnetic_flux , unsigned long long> boost::units::literals::operator"" _fWb (   
    unsigned long long x )
```

Definition at line 84 of file literals.h.

5.4.1.449 operator""_g() [1/2]

```
quantity< mass , unsigned long long> boost::units::literals::operator"" _g (   
    unsigned long long x )
```

Definition at line 66 of file literals.h.

5.4.1.450 operator""_g() [2/2]

```
quantity< mass , long double> boost::units::literals::operator"" _g (   
    long double x )
```

Definition at line 66 of file literals.h.

5.4.1.451 operator""_GA() [1/2]

```
quantity< current , long double> boost::units::literals::operator"" _GA (   
    long double x )
```

Definition at line 68 of file literals.h.

5.4.1.452 operator""_GA() [2/2]

```
quantity< current , unsigned long long> boost::units::literals::operator"" _GA (   
    unsigned long long x )
```

Definition at line 68 of file literals.h.

5.4.1.453 operator""_GBq() [1/2]

```
quantity< activity , long double> boost::units::literals::operator""_GBq ( long double x )
```

Definition at line 90 of file literals.h.

5.4.1.454 operator""_GBq() [2/2]

```
quantity< activity , unsigned long long> boost::units::literals::operator""_GBq ( unsigned long long x )
```

Definition at line 90 of file literals.h.

5.4.1.455 operator""_GC() [1/2]

```
quantity< electric_charge , long double> boost::units::literals::operator""_GC ( long double x )
```

Definition at line 79 of file literals.h.

5.4.1.456 operator""_GC() [2/2]

```
quantity< electric_charge , unsigned long long> boost::units::literals::operator""_GC ( unsigned long long x )
```

Definition at line 79 of file literals.h.

5.4.1.457 operator""_Gcd() [1/2]

```
quantity< luminous_intensity , long double> boost::units::literals::operator""_Gcd ( long double x )
```

Definition at line 71 of file literals.h.

5.4.1.458 operator""_Gcd() [2/2]

```
quantity< luminous_intensity , unsigned long long> boost::units::literals::operator""_Gcd ( unsigned long long x )
```

Definition at line 71 of file literals.h.

5.4.1.459 operator""_Gday() [1/2]

```
quantity< time , long double> boost::units::literals::operator""_Gday ( long double x )
```

Definition at line 96 of file literals.h.

5.4.1.460 operator""_Gday() [2/2]

```
quantity< time , unsigned long long> boost::units::literals::operator"" _Gday (  
    unsigned long long x )
```

Definition at line 96 of file literals.h.

5.4.1.461 operator""_Gdeg() [1/2]

```
quantity< plane_angle , long double> boost::units::literals::operator"" _Gdeg (   
    long double x )
```

Definition at line 97 of file literals.h.

5.4.1.462 operator""_Gdeg() [2/2]

```
quantity< plane_angle , unsigned long long> boost::units::literals::operator"" _Gdeg (   
    unsigned long long x )
```

Definition at line 97 of file literals.h.

5.4.1.463 operator""_GdegC() [1/2]

```
quantity< temperature , long double> boost::units::literals::operator"" _GdegC (   
    long double x )
```

Definition at line 87 of file literals.h.

5.4.1.464 operator""_GdegC() [2/2]

```
quantity< temperature , unsigned long long> boost::units::literals::operator"" _GdegC (   
    unsigned long long x )
```

Definition at line 87 of file literals.h.

5.4.1.465 operator""_GF() [1/2]

```
quantity< capacitance , long double> boost::units::literals::operator"" _GF (   
    long double x )
```

Definition at line 81 of file literals.h.

5.4.1.466 operator""_GF() [2/2]

```
quantity< capacitance , unsigned long long> boost::units::literals::operator"" _GF (   
    unsigned long long x )
```

Definition at line 81 of file literals.h.

5.4.1.467 operator""_Gg() [1/2]

```
quantity< mass , long double> boost::units::literals::operator"" _Gg (
    long double x )
```

Definition at line 66 of file literals.h.

5.4.1.468 operator""_Gg() [2/2]

```
quantity< mass , unsigned long long> boost::units::literals::operator"" _Gg (
    unsigned long long x )
```

Definition at line 66 of file literals.h.

5.4.1.469 operator""_GGy() [1/2]

```
quantity< absorbed_dose , long double> boost::units::literals::operator"" _GGy (
    long double x )
```

Definition at line 91 of file literals.h.

5.4.1.470 operator""_GGy() [2/2]

```
quantity< absorbed_dose , unsigned long long> boost::units::literals::operator"" _GGy (
    unsigned long long x )
```

Definition at line 91 of file literals.h.

5.4.1.471 operator""_GH() [1/2]

```
quantity< inductance , long double> boost::units::literals::operator"" _GH (
    long double x )
```

Definition at line 86 of file literals.h.

5.4.1.472 operator""_GH() [2/2]

```
quantity< inductance , unsigned long long> boost::units::literals::operator"" _GH (
    unsigned long long x )
```

Definition at line 86 of file literals.h.

5.4.1.473 operator""_Gh() [1/2]

```
quantity< time , long double> boost::units::literals::operator"" _Gh (
    long double x )
```

Definition at line 95 of file literals.h.

5.4.1.474 operator""_Gh() [2/2]

```
quantity< time , unsigned long long> boost::units::literals::operator""_Gh (  
    unsigned long long x )
```

Definition at line 95 of file literals.h.

5.4.1.475 operator""_GHz() [1/2]

```
quantity< frequency , long double> boost::units::literals::operator""_GHz (   
    long double x )
```

Definition at line 72 of file literals.h.

5.4.1.476 operator""_GHz() [2/2]

```
quantity< frequency , unsigned long long> boost::units::literals::operator""_GHz (   
    unsigned long long x )
```

Definition at line 72 of file literals.h.

5.4.1.477 operator""_GJ() [1/2]

```
quantity< energy , long double> boost::units::literals::operator""_GJ (   
    long double x )
```

Definition at line 77 of file literals.h.

5.4.1.478 operator""_GJ() [2/2]

```
quantity< energy , unsigned long long> boost::units::literals::operator""_GJ (   
    unsigned long long x )
```

Definition at line 77 of file literals.h.

5.4.1.479 operator""_GK() [1/2]

```
quantity< temperature , long double> boost::units::literals::operator""_GK (   
    long double x )
```

Definition at line 69 of file literals.h.

5.4.1.480 operator""_GK() [2/2]

```
quantity< temperature , unsigned long long> boost::units::literals::operator""_GK (   
    unsigned long long x )
```

Definition at line 69 of file literals.h.

5.4.1.481 operator""_Gkat() [1/2]

```
quantity< catalytic_activity , long double> boost::units::literals::operator"" _Gkat (
    long double x )
```

Definition at line 93 of file literals.h.

5.4.1.482 operator""_Gkat() [2/2]

```
quantity< catalytic_activity , unsigned long long> boost::units::literals::operator"" _Gkat (
    unsigned long long x )
```

Definition at line 93 of file literals.h.

5.4.1.483 operator""_Gl() [1/2]

```
quantity< volume , long double> boost::units::literals::operator"" _Gl (
    long double x )
```

Definition at line 98 of file literals.h.

5.4.1.484 operator""_Gl() [2/2]

```
quantity< volume , unsigned long long> boost::units::literals::operator"" _Gl (
    unsigned long long x )
```

Definition at line 98 of file literals.h.

5.4.1.485 operator""_GL() [1/2]

```
quantity< volume , long double> boost::units::literals::operator"" _GL (
    long double x )
```

Definition at line 99 of file literals.h.

5.4.1.486 operator""_GL() [2/2]

```
quantity< volume , unsigned long long> boost::units::literals::operator"" _GL (
    unsigned long long x )
```

Definition at line 99 of file literals.h.

5.4.1.487 operator""_Glm() [1/2]

```
quantity< luminous_flux , long double> boost::units::literals::operator"" _Glm (
    long double x )
```

Definition at line 88 of file literals.h.

5.4.1.488 operator""_Glm() [2/2]

```
quantity< luminous_flux , unsigned long long> boost::units::literals::operator""_Glm (  
    unsigned long long x )
```

Definition at line 88 of file literals.h.

5.4.1.489 operator""_Glx() [1/2]

```
quantity< illuminance , long double> boost::units::literals::operator""_Glx (   
    long double x )
```

Definition at line 89 of file literals.h.

5.4.1.490 operator""_Glx() [2/2]

```
quantity< illuminance , unsigned long long> boost::units::literals::operator""_Glx (   
    unsigned long long x )
```

Definition at line 89 of file literals.h.

5.4.1.491 operator""_Gm() [1/2]

```
quantity< length , long double> boost::units::literals::operator""_Gm (   
    long double x )
```

Definition at line 65 of file literals.h.

5.4.1.492 operator""_Gm() [2/2]

```
quantity< length , unsigned long long> boost::units::literals::operator""_Gm (   
    unsigned long long x )
```

Definition at line 65 of file literals.h.

5.4.1.493 operator""_Gmin() [1/2]

```
quantity< time , long double> boost::units::literals::operator""_Gmin (   
    long double x )
```

Definition at line 94 of file literals.h.

5.4.1.494 operator""_Gmin() [2/2]

```
quantity< time , unsigned long long> boost::units::literals::operator""_Gmin (   
    unsigned long long x )
```

Definition at line 94 of file literals.h.

5.4.1.495 operator""_Gmol() [1/2]

```
quantity< amount , long double> boost::units::literals::operator""_Gmol ( long double x )
```

Definition at line 70 of file literals.h.

5.4.1.496 operator""_Gmol() [2/2]

```
quantity< amount , unsigned long long> boost::units::literals::operator""_Gmol ( unsigned long long x )
```

Definition at line 70 of file literals.h.

5.4.1.497 operator""_GN() [1/2]

```
quantity< force , long double> boost::units::literals::operator""_GN ( long double x )
```

Definition at line 75 of file literals.h.

5.4.1.498 operator""_GN() [2/2]

```
quantity< force , unsigned long long> boost::units::literals::operator""_GN ( unsigned long long x )
```

Definition at line 75 of file literals.h.

5.4.1.499 operator""_Gohm() [1/2]

```
quantity< resistance , long double> boost::units::literals::operator""_Gohm ( long double x )
```

Definition at line 82 of file literals.h.

5.4.1.500 operator""_Gohm() [2/2]

```
quantity< resistance , unsigned long long> boost::units::literals::operator""_Gohm ( unsigned long long x )
```

Definition at line 82 of file literals.h.

5.4.1.501 operator""_GPa() [1/2]

```
quantity< pressure , long double> boost::units::literals::operator""_GPa ( long double x )
```

Definition at line 76 of file literals.h.

5.4.1.502 operator""_GPa() [2/2]

```
quantity< pressure , unsigned long long> boost::units::literals::operator"" _GPa ( unsigned long long x )
```

Definition at line 76 of file literals.h.

5.4.1.503 operator""_Grad() [1/2]

```
quantity< plane_angle , long double> boost::units::literals::operator"" _Grad ( long double x )
```

Definition at line 73 of file literals.h.

5.4.1.504 operator""_Grad() [2/2]

```
quantity< plane_angle , unsigned long long> boost::units::literals::operator"" _Grad ( unsigned long long x )
```

Definition at line 73 of file literals.h.

5.4.1.505 operator""_Gs() [1/2]

```
quantity< time , long double> boost::units::literals::operator"" _Gs ( long double x )
```

Definition at line 67 of file literals.h.

5.4.1.506 operator""_Gs() [2/2]

```
quantity< time , unsigned long long> boost::units::literals::operator"" _Gs ( unsigned long long x )
```

Definition at line 67 of file literals.h.

5.4.1.507 operator""_GS() [1/2]

```
quantity< conductance , long double> boost::units::literals::operator"" _GS ( long double x )
```

Definition at line 83 of file literals.h.

5.4.1.508 operator""_GS() [2/2]

```
quantity< conductance , unsigned long long> boost::units::literals::operator"" _GS ( unsigned long long x )
```

Definition at line 83 of file literals.h.

5.4.1.509 operator""_Gsr() [1/2]

```
quantity< solid_angle , long double> boost::units::literals::operator""_Gsr ( long double x )
```

Definition at line 74 of file literals.h.

5.4.1.510 operator""_Gsr() [2/2]

```
quantity< solid_angle , unsigned long long> boost::units::literals::operator""_Gsr ( unsigned long long x )
```

Definition at line 74 of file literals.h.

5.4.1.511 operator""_GSv() [1/2]

```
quantity< dose_equivalent , long double> boost::units::literals::operator""_GSv ( long double x )
```

Definition at line 92 of file literals.h.

5.4.1.512 operator""_GSv() [2/2]

```
quantity< dose_equivalent , unsigned long long> boost::units::literals::operator""_GSv ( unsigned long long x )
```

Definition at line 92 of file literals.h.

5.4.1.513 operator""_GT() [1/2]

```
quantity< magnetic_flux_density , long double> boost::units::literals::operator""_GT ( long double x )
```

Definition at line 85 of file literals.h.

5.4.1.514 operator""_GT() [2/2]

```
quantity< magnetic_flux_density , unsigned long long> boost::units::literals::operator""_GT ( unsigned long long x )
```

Definition at line 85 of file literals.h.

5.4.1.515 operator""_Gt() [1/2]

```
quantity< mass , long double> boost::units::literals::operator""_Gt ( long double x )
```

Definition at line 100 of file literals.h.

5.4.1.516 operator""_Gt() [2/2]

```
quantity< mass , unsigned long long> boost::units::literals::operator""_Gt (  
    unsigned long long x )
```

Definition at line 100 of file literals.h.

5.4.1.517 operator""_GV() [1/2]

```
quantity< electric_potential , long double> boost::units::literals::operator""_GV (   
    long double x )
```

Definition at line 80 of file literals.h.

5.4.1.518 operator""_GV() [2/2]

```
quantity< electric_potential , unsigned long long> boost::units::literals::operator""_GV (   
    unsigned long long x )
```

Definition at line 80 of file literals.h.

5.4.1.519 operator""_GW() [1/2]

```
quantity< power , long double> boost::units::literals::operator""_GW (   
    long double x )
```

Definition at line 78 of file literals.h.

5.4.1.520 operator""_GW() [2/2]

```
quantity< power , unsigned long long> boost::units::literals::operator""_GW (   
    unsigned long long x )
```

Definition at line 78 of file literals.h.

5.4.1.521 operator""_GWb() [1/2]

```
quantity< magnetic_flux , long double> boost::units::literals::operator""_GWb (   
    long double x )
```

Definition at line 84 of file literals.h.

5.4.1.522 operator""_GWb() [2/2]

```
quantity< magnetic_flux , unsigned long long> boost::units::literals::operator""_GWb (   
    unsigned long long x )
```

Definition at line 84 of file literals.h.

5.4.1.523 operator""_Gy() [1/2]

```
quantity< absorbed_dose , long double> boost::units::literals::operator"" _Gy ( long double x )
```

Definition at line 91 of file literals.h.

5.4.1.524 operator""_Gy() [2/2]

```
quantity< absorbed_dose , unsigned long long> boost::units::literals::operator"" _Gy ( unsigned long long x )
```

Definition at line 91 of file literals.h.

5.4.1.525 operator""_H() [1/2]

```
quantity< inductance , long double> boost::units::literals::operator"" _H ( long double x )
```

Definition at line 86 of file literals.h.

5.4.1.526 operator""_H() [2/2]

```
quantity< inductance , unsigned long long> boost::units::literals::operator"" _H ( unsigned long long x )
```

Definition at line 86 of file literals.h.

5.4.1.527 operator""_h() [1/2]

```
quantity< time , long double> boost::units::literals::operator"" _h ( long double x )
```

Definition at line 95 of file literals.h.

5.4.1.528 operator""_h() [2/2]

```
quantity< time , unsigned long long> boost::units::literals::operator"" _h ( unsigned long long x )
```

Definition at line 95 of file literals.h.

5.4.1.529 operator""_hA() [1/2]

```
quantity< current , long double> boost::units::literals::operator"" _hA ( long double x )
```

Definition at line 68 of file literals.h.

5.4.1.530 operator"" _hA() [2/2]

```
quantity< current , unsigned long long> boost::units::literals::operator"" _hA (  
    unsigned long long x )
```

Definition at line 68 of file literals.h.

5.4.1.531 operator"" _hBq() [1/2]

```
quantity< activity , unsigned long long> boost::units::literals::operator"" _hBq (   
    unsigned long long x )
```

Definition at line 90 of file literals.h.

5.4.1.532 operator"" _hBq() [2/2]

```
quantity< activity , long double> boost::units::literals::operator"" _hBq (   
    long double x )
```

Definition at line 90 of file literals.h.

5.4.1.533 operator"" _hC() [1/2]

```
quantity< electric_charge , long double> boost::units::literals::operator"" _hC (   
    long double x )
```

Definition at line 79 of file literals.h.

5.4.1.534 operator"" _hC() [2/2]

```
quantity< electric_charge , unsigned long long> boost::units::literals::operator"" _hC (   
    unsigned long long x )
```

Definition at line 79 of file literals.h.

5.4.1.535 operator"" _hcd() [1/2]

```
quantity< luminous_intensity , long double> boost::units::literals::operator"" _hcd (   
    long double x )
```

Definition at line 71 of file literals.h.

5.4.1.536 operator"" _hcd() [2/2]

```
quantity< luminous_intensity , unsigned long long> boost::units::literals::operator"" _hcd (   
    unsigned long long x )
```

Definition at line 71 of file literals.h.

5.4.1.537 operator""_hday() [1/2]

```
quantity< time , long double> boost::units::literals::operator"" _hday ( long double x )
```

Definition at line 96 of file literals.h.

5.4.1.538 operator""_hday() [2/2]

```
quantity< time , unsigned long long> boost::units::literals::operator"" _hday ( unsigned long long x )
```

Definition at line 96 of file literals.h.

5.4.1.539 operator""_hdeg() [1/2]

```
quantity< plane_angle , long double> boost::units::literals::operator"" _hdeg ( long double x )
```

Definition at line 97 of file literals.h.

5.4.1.540 operator""_hdeg() [2/2]

```
quantity< plane_angle , unsigned long long> boost::units::literals::operator"" _hdeg ( unsigned long long x )
```

Definition at line 97 of file literals.h.

5.4.1.541 operator""_hdegC() [1/2]

```
quantity< temperature , long double> boost::units::literals::operator"" _hdegC ( long double x )
```

Definition at line 87 of file literals.h.

5.4.1.542 operator""_hdegC() [2/2]

```
quantity< temperature , unsigned long long> boost::units::literals::operator"" _hdegC ( unsigned long long x )
```

Definition at line 87 of file literals.h.

5.4.1.543 operator""_hF() [1/2]

```
quantity< capacitance , long double> boost::units::literals::operator"" _hF ( long double x )
```

Definition at line 81 of file literals.h.

5.4.1.544 operator""_hF() [2/2]

```
quantity< capacitance , unsigned long long> boost::units::literals::operator"" _hF ( unsigned long long x )
```

Definition at line 81 of file literals.h.

5.4.1.545 operator""_hg() [1/2]

```
quantity< mass , unsigned long long> boost::units::literals::operator"" _hg ( unsigned long long x )
```

Definition at line 66 of file literals.h.

5.4.1.546 operator""_hg() [2/2]

```
quantity< mass , long double> boost::units::literals::operator"" _hg ( long double x )
```

Definition at line 66 of file literals.h.

5.4.1.547 operator""_hGy() [1/2]

```
quantity< absorbed_dose , long double> boost::units::literals::operator"" _hGy ( long double x )
```

Definition at line 91 of file literals.h.

5.4.1.548 operator""_hGy() [2/2]

```
quantity< absorbed_dose , unsigned long long> boost::units::literals::operator"" _hGy ( unsigned long long x )
```

Definition at line 91 of file literals.h.

5.4.1.549 operator""_hH() [1/2]

```
quantity< inductance , unsigned long long> boost::units::literals::operator"" _hH ( unsigned long long x )
```

Definition at line 86 of file literals.h.

5.4.1.550 operator""_hH() [2/2]

```
quantity< inductance , long double> boost::units::literals::operator"" _hH ( long double x )
```

Definition at line 86 of file literals.h.

5.4.1.551 operator""_hh() [1/2]

```
quantity< time , long double> boost::units::literals::operator"" _hh ( long double x )
```

Definition at line 95 of file literals.h.

5.4.1.552 operator""_hh() [2/2]

```
quantity< time , unsigned long long> boost::units::literals::operator"" _hh ( unsigned long long x )
```

Definition at line 95 of file literals.h.

5.4.1.553 operator""_hHz() [1/2]

```
quantity< frequency , long double> boost::units::literals::operator"" _hHz ( long double x )
```

Definition at line 72 of file literals.h.

5.4.1.554 operator""_hHz() [2/2]

```
quantity< frequency , unsigned long long> boost::units::literals::operator"" _hHz ( unsigned long long x )
```

Definition at line 72 of file literals.h.

5.4.1.555 operator""_hJ() [1/2]

```
quantity< energy , long double> boost::units::literals::operator"" _hJ ( long double x )
```

Definition at line 77 of file literals.h.

5.4.1.556 operator""_hJ() [2/2]

```
quantity< energy , unsigned long long> boost::units::literals::operator"" _hJ ( unsigned long long x )
```

Definition at line 77 of file literals.h.

5.4.1.557 operator""_hK() [1/2]

```
quantity< temperature , long double> boost::units::literals::operator"" _hK ( long double x )
```

Definition at line 69 of file literals.h.

5.4.1.558 operator"" _hK() [2/2]

```
quantity< temperature , unsigned long long> boost::units::literals::operator"" _hK ( unsigned long long x )
```

Definition at line 69 of file literals.h.

5.4.1.559 operator"" _hkat() [1/2]

```
quantity< catalytic_activity , long double> boost::units::literals::operator"" _hkat ( long double x )
```

Definition at line 93 of file literals.h.

5.4.1.560 operator"" _hkat() [2/2]

```
quantity< catalytic_activity , unsigned long long> boost::units::literals::operator"" _hkat ( unsigned long long x )
```

Definition at line 93 of file literals.h.

5.4.1.561 operator"" _hl() [1/2]

```
quantity< volume , unsigned long long> boost::units::literals::operator"" _hl ( unsigned long long x )
```

Definition at line 98 of file literals.h.

5.4.1.562 operator"" _hl() [2/2]

```
quantity< volume , long double> boost::units::literals::operator"" _hl ( long double x )
```

Definition at line 98 of file literals.h.

5.4.1.563 operator"" _hL() [1/2]

```
quantity< volume , unsigned long long> boost::units::literals::operator"" _hL ( unsigned long long x )
```

Definition at line 99 of file literals.h.

5.4.1.564 operator"" _hL() [2/2]

```
quantity< volume , long double> boost::units::literals::operator"" _hL ( long double x )
```

Definition at line 99 of file literals.h.

5.4.1.565 operator""_hlm() [1/2]

```
quantity< luminous_flux , long double> boost::units::literals::operator"" _hlm ( long double x )
```

Definition at line 88 of file literals.h.

5.4.1.566 operator""_hlm() [2/2]

```
quantity< luminous_flux , unsigned long long> boost::units::literals::operator"" _hlm ( unsigned long long x )
```

Definition at line 88 of file literals.h.

5.4.1.567 operator""_hlx() [1/2]

```
quantity< illuminance , long double> boost::units::literals::operator"" _hlx ( long double x )
```

Definition at line 89 of file literals.h.

5.4.1.568 operator""_hlx() [2/2]

```
quantity< illuminance , unsigned long long> boost::units::literals::operator"" _hlx ( unsigned long long x )
```

Definition at line 89 of file literals.h.

5.4.1.569 operator""_hm() [1/2]

```
quantity< length , long double> boost::units::literals::operator"" _hm ( long double x )
```

Definition at line 65 of file literals.h.

5.4.1.570 operator""_hm() [2/2]

```
quantity< length , unsigned long long> boost::units::literals::operator"" _hm ( unsigned long long x )
```

Definition at line 65 of file literals.h.

5.4.1.571 operator""_hmin() [1/2]

```
quantity< time , unsigned long long> boost::units::literals::operator"" _hmin ( unsigned long long x )
```

Definition at line 94 of file literals.h.

5.4.1.572 operator"" _hmin() [2/2]

```
quantity< time , long double> boost::units::literals::operator"" _hmin ( long double x )
```

Definition at line 94 of file literals.h.

5.4.1.573 operator"" _hmol() [1/2]

```
quantity< amount , unsigned long long> boost::units::literals::operator"" _hmol ( unsigned long long x )
```

Definition at line 70 of file literals.h.

5.4.1.574 operator"" _hmol() [2/2]

```
quantity< amount , long double> boost::units::literals::operator"" _hmol ( long double x )
```

Definition at line 70 of file literals.h.

5.4.1.575 operator"" _hN() [1/2]

```
quantity< force , long double> boost::units::literals::operator"" _hN ( long double x )
```

Definition at line 75 of file literals.h.

5.4.1.576 operator"" _hN() [2/2]

```
quantity< force , unsigned long long> boost::units::literals::operator"" _hN ( unsigned long long x )
```

Definition at line 75 of file literals.h.

5.4.1.577 operator"" _hohm() [1/2]

```
quantity< resistance , long double> boost::units::literals::operator"" _hohm ( long double x )
```

Definition at line 82 of file literals.h.

5.4.1.578 operator"" _hohm() [2/2]

```
quantity< resistance , unsigned long long> boost::units::literals::operator"" _hohm ( unsigned long long x )
```

Definition at line 82 of file literals.h.

5.4.1.579 operator""_hPa() [1/2]

```
quantity< pressure , long double> boost::units::literals::operator"" _hPa ( long double x )
```

Definition at line 76 of file literals.h.

5.4.1.580 operator""_hPa() [2/2]

```
quantity< pressure , unsigned long long> boost::units::literals::operator"" _hPa ( unsigned long long x )
```

Definition at line 76 of file literals.h.

5.4.1.581 operator""_hrad() [1/2]

```
quantity< plane_angle , long double> boost::units::literals::operator"" _hrad ( long double x )
```

Definition at line 73 of file literals.h.

5.4.1.582 operator""_hrad() [2/2]

```
quantity< plane_angle , unsigned long long> boost::units::literals::operator"" _hrad ( unsigned long long x )
```

Definition at line 73 of file literals.h.

5.4.1.583 operator""_hs() [1/2]

```
quantity< time , long double> boost::units::literals::operator"" _hs ( long double x )
```

Definition at line 67 of file literals.h.

5.4.1.584 operator""_hs() [2/2]

```
quantity< time , unsigned long long> boost::units::literals::operator"" _hs ( unsigned long long x )
```

Definition at line 67 of file literals.h.

5.4.1.585 operator""_hS() [1/2]

```
quantity< conductance , long double> boost::units::literals::operator"" _hS ( long double x )
```

Definition at line 83 of file literals.h.

5.4.1.586 operator""_hS() [2/2]

```
quantity< conductance , unsigned long long> boost::units::literals::operator"" _hS ( unsigned long long x )
```

Definition at line 83 of file literals.h.

5.4.1.587 operator""_hsr() [1/2]

```
quantity< solid_angle , unsigned long long> boost::units::literals::operator"" _hsr ( unsigned long long x )
```

Definition at line 74 of file literals.h.

5.4.1.588 operator""_hsr() [2/2]

```
quantity< solid_angle , long double> boost::units::literals::operator"" _hsr ( long double x )
```

Definition at line 74 of file literals.h.

5.4.1.589 operator""_hSv() [1/2]

```
quantity< dose_equivalent , long double> boost::units::literals::operator"" _hSv ( long double x )
```

Definition at line 92 of file literals.h.

5.4.1.590 operator""_hSv() [2/2]

```
quantity< dose_equivalent , unsigned long long> boost::units::literals::operator"" _hSv ( unsigned long long x )
```

Definition at line 92 of file literals.h.

5.4.1.591 operator""_hT() [1/2]

```
quantity< magnetic_flux_density , long double> boost::units::literals::operator"" _hT ( long double x )
```

Definition at line 85 of file literals.h.

5.4.1.592 operator""_hT() [2/2]

```
quantity< magnetic_flux_density , unsigned long long> boost::units::literals::operator"" _hT ( unsigned long long x )
```

Definition at line 85 of file literals.h.

5.4.1.593 operator""_ht() [1/2]

```
quantity< mass , long double> boost::units::literals::operator"" _ht ( long double x )
```

Definition at line 100 of file literals.h.

5.4.1.594 operator""_ht() [2/2]

```
quantity< mass , unsigned long long> boost::units::literals::operator"" _ht ( unsigned long long x )
```

Definition at line 100 of file literals.h.

5.4.1.595 operator""_hV() [1/2]

```
quantity< electric_potential , long double> boost::units::literals::operator"" _hV ( long double x )
```

Definition at line 80 of file literals.h.

5.4.1.596 operator""_hV() [2/2]

```
quantity< electric_potential , unsigned long long> boost::units::literals::operator"" _hV ( unsigned long long x )
```

Definition at line 80 of file literals.h.

5.4.1.597 operator""_hW() [1/2]

```
quantity< power , unsigned long long> boost::units::literals::operator"" _hW ( unsigned long long x )
```

Definition at line 78 of file literals.h.

5.4.1.598 operator""_hW() [2/2]

```
quantity< power , long double> boost::units::literals::operator"" _hW ( long double x )
```

Definition at line 78 of file literals.h.

5.4.1.599 operator""_hWb() [1/2]

```
quantity< magnetic_flux , long double> boost::units::literals::operator"" _hWb ( long double x )
```

Definition at line 84 of file literals.h.

5.4.1.600 operator""_hWb() [2/2]

```
quantity< magnetic_flux , unsigned long long> boost::units::literals::operator""_hWb ( unsigned long long x )
```

Definition at line 84 of file literals.h.

5.4.1.601 operator""_Hz() [1/2]

```
quantity< frequency , long double> boost::units::literals::operator""_Hz ( long double x )
```

Definition at line 72 of file literals.h.

5.4.1.602 operator""_Hz() [2/2]

```
quantity< frequency , unsigned long long> boost::units::literals::operator""_Hz ( unsigned long long x )
```

Definition at line 72 of file literals.h.

5.4.1.603 operator""_J() [1/2]

```
quantity< energy , long double> boost::units::literals::operator""_J ( long double x )
```

Definition at line 77 of file literals.h.

5.4.1.604 operator""_J() [2/2]

```
quantity< energy , unsigned long long> boost::units::literals::operator""_J ( unsigned long long x )
```

Definition at line 77 of file literals.h.

5.4.1.605 operator""_K() [1/2]

```
quantity< temperature , long double> boost::units::literals::operator""_K ( long double x )
```

Definition at line 69 of file literals.h.

5.4.1.606 operator""_K() [2/2]

```
quantity< temperature , unsigned long long> boost::units::literals::operator""_K ( unsigned long long x )
```

Definition at line 69 of file literals.h.

5.4.1.607 operator""_kA() [1/2]

```
quantity< current , long double> boost::units::literals::operator"" _kA ( long double x )
```

Definition at line 68 of file literals.h.

5.4.1.608 operator""_kA() [2/2]

```
quantity< current , unsigned long long> boost::units::literals::operator"" _kA ( unsigned long long x )
```

Definition at line 68 of file literals.h.

5.4.1.609 operator""_kat() [1/2]

```
quantity< catalytic_activity , long double> boost::units::literals::operator"" _kat ( long double x )
```

Definition at line 93 of file literals.h.

5.4.1.610 operator""_kat() [2/2]

```
quantity< catalytic_activity , unsigned long long> boost::units::literals::operator"" _kat ( unsigned long long x )
```

Definition at line 93 of file literals.h.

5.4.1.611 operator""_kBq() [1/2]

```
quantity< activity , long double> boost::units::literals::operator"" _kBq ( long double x )
```

Definition at line 90 of file literals.h.

5.4.1.612 operator""_kBq() [2/2]

```
quantity< activity , unsigned long long> boost::units::literals::operator"" _kBq ( unsigned long long x )
```

Definition at line 90 of file literals.h.

5.4.1.613 operator""_kC() [1/2]

```
quantity< electric_charge , unsigned long long> boost::units::literals::operator"" _kC ( unsigned long long x )
```

Definition at line 79 of file literals.h.

5.4.1.614 operator""_kC() [2/2]

```
quantity< electric_charge , long double> boost::units::literals::operator"" _kC (
    long double x )
```

Definition at line 79 of file literals.h.

5.4.1.615 operator""_kcd() [1/2]

```
quantity< luminous_intensity , unsigned long long> boost::units::literals::operator"" _kcd (
    unsigned long long x )
```

Definition at line 71 of file literals.h.

5.4.1.616 operator""_kcd() [2/2]

```
quantity< luminous_intensity , long double> boost::units::literals::operator"" _kcd (
    long double x )
```

Definition at line 71 of file literals.h.

5.4.1.617 operator""_kday() [1/2]

```
quantity< time , unsigned long long> boost::units::literals::operator"" _kday (
    unsigned long long x )
```

Definition at line 96 of file literals.h.

5.4.1.618 operator""_kday() [2/2]

```
quantity< time , long double> boost::units::literals::operator"" _kday (
    long double x )
```

Definition at line 96 of file literals.h.

5.4.1.619 operator""_kdeg() [1/2]

```
quantity< plane_angle , long double> boost::units::literals::operator"" _kdeg (
    long double x )
```

Definition at line 97 of file literals.h.

5.4.1.620 operator""_kdeg() [2/2]

```
quantity< plane_angle , unsigned long long> boost::units::literals::operator"" _kdeg (
    unsigned long long x )
```

Definition at line 97 of file literals.h.

5.4.1.621 operator""_kdegC() [1/2]

```
quantity< temperature , long double> boost::units::literals::operator"" _kdegC (
    long double x )
```

Definition at line 87 of file literals.h.

5.4.1.622 operator""_kdegC() [2/2]

```
quantity< temperature , unsigned long long> boost::units::literals::operator"" _kdegC (
    unsigned long long x )
```

Definition at line 87 of file literals.h.

5.4.1.623 operator""_kF() [1/2]

```
quantity< capacitance , long double> boost::units::literals::operator"" _kF (
    long double x )
```

Definition at line 81 of file literals.h.

5.4.1.624 operator""_kF() [2/2]

```
quantity< capacitance , unsigned long long> boost::units::literals::operator"" _kF (
    unsigned long long x )
```

Definition at line 81 of file literals.h.

5.4.1.625 operator""_kg() [1/2]

```
quantity< mass , long double> boost::units::literals::operator"" _kg (
    long double x )
```

Definition at line 66 of file literals.h.

5.4.1.626 operator""_kg() [2/2]

```
quantity< mass , unsigned long long> boost::units::literals::operator"" _kg (
    unsigned long long x )
```

Definition at line 66 of file literals.h.

5.4.1.627 operator""_kGy() [1/2]

```
quantity< absorbed_dose , long double> boost::units::literals::operator"" _kGy (
    long double x )
```

Definition at line 91 of file literals.h.

5.4.1.628 operator""_kGy() [2/2]

```
quantity< absorbed_dose , unsigned long long> boost::units::literals::operator""_kGy ( unsigned long long x )
```

Definition at line 91 of file literals.h.

5.4.1.629 operator""_kH() [1/2]

```
quantity< inductance , long double> boost::units::literals::operator""_kH ( long double x )
```

Definition at line 86 of file literals.h.

5.4.1.630 operator""_kH() [2/2]

```
quantity< inductance , unsigned long long> boost::units::literals::operator""_kH ( unsigned long long x )
```

Definition at line 86 of file literals.h.

5.4.1.631 operator""_kh() [1/2]

```
quantity< time , long double> boost::units::literals::operator""_kh ( long double x )
```

Definition at line 95 of file literals.h.

5.4.1.632 operator""_kh() [2/2]

```
quantity< time , unsigned long long> boost::units::literals::operator""_kh ( unsigned long long x )
```

Definition at line 95 of file literals.h.

5.4.1.633 operator""_kHz() [1/2]

```
quantity< frequency , long double> boost::units::literals::operator""_kHz ( long double x )
```

Definition at line 72 of file literals.h.

5.4.1.634 operator""_kHz() [2/2]

```
quantity< frequency , unsigned long long> boost::units::literals::operator""_kHz ( unsigned long long x )
```

Definition at line 72 of file literals.h.

5.4.1.635 operator""_kJ() [1/2]

```
quantity< energy , long double> boost::units::literals::operator"" _kJ ( long double x )
```

Definition at line 77 of file literals.h.

5.4.1.636 operator""_kJ() [2/2]

```
quantity< energy , unsigned long long> boost::units::literals::operator"" _kJ ( unsigned long long x )
```

Definition at line 77 of file literals.h.

5.4.1.637 operator""_kK() [1/2]

```
quantity< temperature , long double> boost::units::literals::operator"" _kK ( long double x )
```

Definition at line 69 of file literals.h.

5.4.1.638 operator""_kK() [2/2]

```
quantity< temperature , unsigned long long> boost::units::literals::operator"" _kK ( unsigned long long x )
```

Definition at line 69 of file literals.h.

5.4.1.639 operator""_kkat() [1/2]

```
quantity< catalytic_activity , long double> boost::units::literals::operator"" _kkat ( long double x )
```

Definition at line 93 of file literals.h.

5.4.1.640 operator""_kkat() [2/2]

```
quantity< catalytic_activity , unsigned long long> boost::units::literals::operator"" _kkat ( unsigned long long x )
```

Definition at line 93 of file literals.h.

5.4.1.641 operator""_kl() [1/2]

```
quantity< volume , long double> boost::units::literals::operator"" _kl ( long double x )
```

Definition at line 98 of file literals.h.

5.4.1.642 operator""_kl() [2/2]

```
quantity< volume , unsigned long long> boost::units::literals::operator""_kl (  
    unsigned long long x )
```

Definition at line 98 of file literals.h.

5.4.1.643 operator""_kL() [1/2]

```
quantity< volume , long double> boost::units::literals::operator""_kL (   
    long double x )
```

Definition at line 99 of file literals.h.

5.4.1.644 operator""_kL() [2/2]

```
quantity< volume , unsigned long long> boost::units::literals::operator""_kL (   
    unsigned long long x )
```

Definition at line 99 of file literals.h.

5.4.1.645 operator""_klm() [1/2]

```
quantity< luminous_flux , long double> boost::units::literals::operator""_klm (   
    long double x )
```

Definition at line 88 of file literals.h.

5.4.1.646 operator""_klm() [2/2]

```
quantity< luminous_flux , unsigned long long> boost::units::literals::operator""_klm (   
    unsigned long long x )
```

Definition at line 88 of file literals.h.

5.4.1.647 operator""_klx() [1/2]

```
quantity< illuminance , unsigned long long> boost::units::literals::operator""_klx (   
    unsigned long long x )
```

Definition at line 89 of file literals.h.

5.4.1.648 operator""_klx() [2/2]

```
quantity< illuminance , long double> boost::units::literals::operator""_klx (   
    long double x )
```

Definition at line 89 of file literals.h.

5.4.1.649 operator""_km() [1/2]

```
quantity< length , unsigned long long> boost::units::literals::operator"" _km (
    unsigned long long x )
```

Definition at line 65 of file literals.h.

5.4.1.650 operator""_km() [2/2]

```
quantity< length , long double> boost::units::literals::operator"" _km (
    long double x )
```

Definition at line 65 of file literals.h.

5.4.1.651 operator""_kmin() [1/2]

```
quantity< time , long double> boost::units::literals::operator"" _kmin (
    long double x )
```

Definition at line 94 of file literals.h.

5.4.1.652 operator""_kmin() [2/2]

```
quantity< time , unsigned long long> boost::units::literals::operator"" _kmin (
    unsigned long long x )
```

Definition at line 94 of file literals.h.

5.4.1.653 operator""_kmol() [1/2]

```
quantity< amount , long double> boost::units::literals::operator"" _kmol (
    long double x )
```

Definition at line 70 of file literals.h.

5.4.1.654 operator""_kmol() [2/2]

```
quantity< amount , unsigned long long> boost::units::literals::operator"" _kmol (
    unsigned long long x )
```

Definition at line 70 of file literals.h.

5.4.1.655 operator""_kN() [1/2]

```
quantity< force , long double> boost::units::literals::operator"" _kN (
    long double x )
```

Definition at line 75 of file literals.h.

5.4.1.656 operator""_kN() [2/2]

```
quantity< force , unsigned long long> boost::units::literals::operator"" _kN ( unsigned long long x )
```

Definition at line 75 of file literals.h.

5.4.1.657 operator""_kohm() [1/2]

```
quantity< resistance , long double> boost::units::literals::operator"" _kohm ( long double x )
```

Definition at line 82 of file literals.h.

5.4.1.658 operator""_kohm() [2/2]

```
quantity< resistance , unsigned long long> boost::units::literals::operator"" _kohm ( unsigned long long x )
```

Definition at line 82 of file literals.h.

5.4.1.659 operator""_kPa() [1/2]

```
quantity< pressure , long double> boost::units::literals::operator"" _kPa ( long double x )
```

Definition at line 76 of file literals.h.

5.4.1.660 operator""_kPa() [2/2]

```
quantity< pressure , unsigned long long> boost::units::literals::operator"" _kPa ( unsigned long long x )
```

Definition at line 76 of file literals.h.

5.4.1.661 operator""_krad() [1/2]

```
quantity< plane_angle , long double> boost::units::literals::operator"" _krad ( long double x )
```

Definition at line 73 of file literals.h.

5.4.1.662 operator""_krad() [2/2]

```
quantity< plane_angle , unsigned long long> boost::units::literals::operator"" _krad ( unsigned long long x )
```

Definition at line 73 of file literals.h.

5.4.1.663 operator""_ks() [1/2]

```
quantity< time , long double> boost::units::literals::operator""_ks (  
    long double x )
```

Definition at line 67 of file literals.h.

5.4.1.664 operator""_ks() [2/2]

```
quantity< time , unsigned long long> boost::units::literals::operator""_ks (   
    unsigned long long x )
```

Definition at line 67 of file literals.h.

5.4.1.665 operator""_kS() [1/2]

```
quantity< conductance , long double> boost::units::literals::operator""_kS (   
    long double x )
```

Definition at line 83 of file literals.h.

5.4.1.666 operator""_kS() [2/2]

```
quantity< conductance , unsigned long long> boost::units::literals::operator""_kS (   
    unsigned long long x )
```

Definition at line 83 of file literals.h.

5.4.1.667 operator""_ksr() [1/2]

```
quantity< solid_angle , long double> boost::units::literals::operator""_ksr (   
    long double x )
```

Definition at line 74 of file literals.h.

5.4.1.668 operator""_ksr() [2/2]

```
quantity< solid_angle , unsigned long long> boost::units::literals::operator""_ksr (   
    unsigned long long x )
```

Definition at line 74 of file literals.h.

5.4.1.669 operator""_kSv() [1/2]

```
quantity< dose_equivalent , long double> boost::units::literals::operator""_kSv (   
    long double x )
```

Definition at line 92 of file literals.h.

5.4.1.670 operator"" _kSv() [2/2]

```
quantity< dose_equivalent , unsigned long long> boost::units::literals::operator"" _kSv ( unsigned long long x )
```

Definition at line 92 of file literals.h.

5.4.1.671 operator"" _kT() [1/2]

```
quantity< magnetic_flux_density , long double> boost::units::literals::operator"" _kT ( long double x )
```

Definition at line 85 of file literals.h.

5.4.1.672 operator"" _kT() [2/2]

```
quantity< magnetic_flux_density , unsigned long long> boost::units::literals::operator"" _kT ( unsigned long long x )
```

Definition at line 85 of file literals.h.

5.4.1.673 operator"" _kt() [1/2]

```
quantity< mass , unsigned long long> boost::units::literals::operator"" _kt ( unsigned long long x )
```

Definition at line 100 of file literals.h.

5.4.1.674 operator"" _kt() [2/2]

```
quantity< mass , long double> boost::units::literals::operator"" _kt ( long double x )
```

Definition at line 100 of file literals.h.

5.4.1.675 operator"" _kV() [1/2]

```
quantity< electric_potential , long double> boost::units::literals::operator"" _kV ( long double x )
```

Definition at line 80 of file literals.h.

5.4.1.676 operator"" _kV() [2/2]

```
quantity< electric_potential , unsigned long long> boost::units::literals::operator"" _kV ( unsigned long long x )
```

Definition at line 80 of file literals.h.

5.4.1.677 operator""_kW() [1/2]

```
quantity< power , long double> boost::units::literals::operator"" _kW (
    long double x )
```

Definition at line 78 of file literals.h.

5.4.1.678 operator""_kW() [2/2]

```
quantity< power , unsigned long long> boost::units::literals::operator"" _kW (
    unsigned long long x )
```

Definition at line 78 of file literals.h.

5.4.1.679 operator""_kWb() [1/2]

```
quantity< magnetic_flux , long double> boost::units::literals::operator"" _kWb (
    long double x )
```

Definition at line 84 of file literals.h.

5.4.1.680 operator""_kWb() [2/2]

```
quantity< magnetic_flux , unsigned long long> boost::units::literals::operator"" _kWb (
    unsigned long long x )
```

Definition at line 84 of file literals.h.

5.4.1.681 operator""_l() [1/2]

```
quantity< volume , long double> boost::units::literals::operator"" _l (
    long double x )
```

Definition at line 98 of file literals.h.

5.4.1.682 operator""_l() [2/2]

```
quantity< volume , unsigned long long> boost::units::literals::operator"" _l (
    unsigned long long x )
```

Definition at line 98 of file literals.h.

5.4.1.683 operator""_L() [1/2]

```
quantity< volume , unsigned long long> boost::units::literals::operator"" _L (
    unsigned long long x )
```

Definition at line 99 of file literals.h.

5.4.1.684 operator""_L() [2/2]

```
quantity< volume , long double> boost::units::literals::operator""_L ( long double x )
```

Definition at line 99 of file literals.h.

5.4.1.685 operator""_lm() [1/2]

```
quantity< luminous_flux , long double> boost::units::literals::operator""_lm ( long double x )
```

Definition at line 88 of file literals.h.

5.4.1.686 operator""_lm() [2/2]

```
quantity< luminous_flux , unsigned long long> boost::units::literals::operator""_lm ( unsigned long long x )
```

Definition at line 88 of file literals.h.

5.4.1.687 operator""_lx() [1/2]

```
quantity< illuminance , long double> boost::units::literals::operator""_lx ( long double x )
```

Definition at line 89 of file literals.h.

5.4.1.688 operator""_lx() [2/2]

```
quantity< illuminance , unsigned long long> boost::units::literals::operator""_lx ( unsigned long long x )
```

Definition at line 89 of file literals.h.

5.4.1.689 operator""_m() [1/2]

```
quantity< length , long double> boost::units::literals::operator""_m ( long double x )
```

Definition at line 65 of file literals.h.

5.4.1.690 operator""_m() [2/2]

```
quantity< length , unsigned long long> boost::units::literals::operator""_m ( unsigned long long x )
```

Definition at line 65 of file literals.h.

5.4.1.691 operator""_MA() [1/2]

```
quantity< current , unsigned long long> boost::units::literals::operator"" _MA (
    unsigned long long x )
```

Definition at line 68 of file literals.h.

5.4.1.692 operator""_mA() [1/2]

```
quantity< current , long double> boost::units::literals::operator"" _mA (
    long double x )
```

Definition at line 68 of file literals.h.

5.4.1.693 operator""_mA() [2/2]

```
quantity< current , unsigned long long> boost::units::literals::operator"" _mA (
    unsigned long long x )
```

Definition at line 68 of file literals.h.

5.4.1.694 operator""_MA() [2/2]

```
quantity< current , long double> boost::units::literals::operator"" _MA (
    long double x )
```

Definition at line 68 of file literals.h.

5.4.1.695 operator""_MBq() [1/2]

```
quantity< activity , long double> boost::units::literals::operator"" _MBq (
    long double x )
```

Definition at line 90 of file literals.h.

5.4.1.696 operator""_MBq() [2/2]

```
quantity< activity , unsigned long long> boost::units::literals::operator"" _MBq (
    unsigned long long x )
```

Definition at line 90 of file literals.h.

5.4.1.697 operator""_mBq() [1/2]

```
quantity< activity , long double> boost::units::literals::operator"" _mBq (
    long double x )
```

Definition at line 90 of file literals.h.

5.4.1.698 operator""_mBq() [2/2]

```
quantity< activity , unsigned long long> boost::units::literals::operator"" _mBq ( unsigned long long x )
```

Definition at line 90 of file literals.h.

5.4.1.699 operator""_mC() [1/2]

```
quantity< electric_charge , long double> boost::units::literals::operator"" _mC ( long double x )
```

Definition at line 79 of file literals.h.

5.4.1.700 operator""_MC() [1/2]

```
quantity< electric_charge , long double> boost::units::literals::operator"" _MC ( long double x )
```

Definition at line 79 of file literals.h.

5.4.1.701 operator""_MC() [2/2]

```
quantity< electric_charge , unsigned long long> boost::units::literals::operator"" _MC ( unsigned long long x )
```

Definition at line 79 of file literals.h.

5.4.1.702 operator""_mC() [2/2]

```
quantity< electric_charge , unsigned long long> boost::units::literals::operator"" _mC ( unsigned long long x )
```

Definition at line 79 of file literals.h.

5.4.1.703 operator""_Mcd() [1/2]

```
quantity< luminous_intensity , long double> boost::units::literals::operator"" _Mcd ( long double x )
```

Definition at line 71 of file literals.h.

5.4.1.704 operator""_Mcd() [2/2]

```
quantity< luminous_intensity , unsigned long long> boost::units::literals::operator"" _Mcd ( unsigned long long x )
```

Definition at line 71 of file literals.h.

5.4.1.705 operator""_mcd() [1/2]

```
quantity< luminous_intensity , long double> boost::units::literals::operator"" _mcd (
    long double x )
```

Definition at line 71 of file literals.h.

5.4.1.706 operator""_mcd() [2/2]

```
quantity< luminous_intensity , unsigned long long> boost::units::literals::operator"" _mcd (
    unsigned long long x )
```

Definition at line 71 of file literals.h.

5.4.1.707 operator""_Mday() [1/2]

```
quantity< time , unsigned long long> boost::units::literals::operator"" _Mday (
    unsigned long long x )
```

Definition at line 96 of file literals.h.

5.4.1.708 operator""_mday() [1/2]

```
quantity< time , long double> boost::units::literals::operator"" _mday (
    long double x )
```

Definition at line 96 of file literals.h.

5.4.1.709 operator""_mday() [2/2]

```
quantity< time , unsigned long long> boost::units::literals::operator"" _mday (
    unsigned long long x )
```

Definition at line 96 of file literals.h.

5.4.1.710 operator""_Mday() [2/2]

```
quantity< time , long double> boost::units::literals::operator"" _Mday (
    long double x )
```

Definition at line 96 of file literals.h.

5.4.1.711 operator""_Mdeg() [1/2]

```
quantity< plane_angle , unsigned long long> boost::units::literals::operator"" _Mdeg (
    unsigned long long x )
```

Definition at line 97 of file literals.h.

5.4.1.712 operator""_Mdeg() [2/2]

```
quantity< plane_angle , long double> boost::units::literals::operator"" _Mdeg ( long double x )
```

Definition at line 97 of file literals.h.

5.4.1.713 operator""_mdeg() [1/2]

```
quantity< plane_angle , long double> boost::units::literals::operator"" _mdeg ( long double x )
```

Definition at line 97 of file literals.h.

5.4.1.714 operator""_mdeg() [2/2]

```
quantity< plane_angle , unsigned long long> boost::units::literals::operator"" _mdeg ( unsigned long long x )
```

Definition at line 97 of file literals.h.

5.4.1.715 operator""_MdegC() [1/2]

```
quantity< temperature , long double> boost::units::literals::operator"" _MdegC ( long double x )
```

Definition at line 87 of file literals.h.

5.4.1.716 operator""_MdegC() [2/2]

```
quantity< temperature , unsigned long long> boost::units::literals::operator"" _MdegC ( unsigned long long x )
```

Definition at line 87 of file literals.h.

5.4.1.717 operator""_mdegC() [1/2]

```
quantity< temperature , long double> boost::units::literals::operator"" _mdegC ( long double x )
```

Definition at line 87 of file literals.h.

5.4.1.718 operator""_mdegC() [2/2]

```
quantity< temperature , unsigned long long> boost::units::literals::operator"" _mdegC ( unsigned long long x )
```

Definition at line 87 of file literals.h.

5.4.1.719 operator""_mF() [1/2]

```
quantity< capacitance , long double> boost::units::literals::operator"" _mF ( long double x )
```

Definition at line 81 of file literals.h.

5.4.1.720 operator""_MF() [1/2]

```
quantity< capacitance , long double> boost::units::literals::operator"" _MF ( long double x )
```

Definition at line 81 of file literals.h.

5.4.1.721 operator""_MF() [2/2]

```
quantity< capacitance , unsigned long long> boost::units::literals::operator"" _MF ( unsigned long long x )
```

Definition at line 81 of file literals.h.

5.4.1.722 operator""_mF() [2/2]

```
quantity< capacitance , unsigned long long> boost::units::literals::operator"" _mF ( unsigned long long x )
```

Definition at line 81 of file literals.h.

5.4.1.723 operator""_mg() [1/2]

```
quantity< mass , long double> boost::units::literals::operator"" _mg ( long double x )
```

Definition at line 66 of file literals.h.

5.4.1.724 operator""_mg() [2/2]

```
quantity< mass , unsigned long long> boost::units::literals::operator"" _mg ( unsigned long long x )
```

Definition at line 66 of file literals.h.

5.4.1.725 operator""_Mg() [1/2]

```
quantity< mass , long double> boost::units::literals::operator"" _Mg ( long double x )
```

Definition at line 66 of file literals.h.

5.4.1.726 operator""_Mg() [2/2]

```
quantity< mass , unsigned long long> boost::units::literals::operator"" _Mg (  
    unsigned long long x )
```

Definition at line 66 of file literals.h.

5.4.1.727 operator""_MGy() [1/2]

```
quantity< absorbed_dose , unsigned long long> boost::units::literals::operator"" _MGy (   
    unsigned long long x )
```

Definition at line 91 of file literals.h.

5.4.1.728 operator""_mGy() [1/2]

```
quantity< absorbed_dose , long double> boost::units::literals::operator"" _mGy (   
    long double x )
```

Definition at line 91 of file literals.h.

5.4.1.729 operator""_mGy() [2/2]

```
quantity< absorbed_dose , unsigned long long> boost::units::literals::operator"" _mGy (   
    unsigned long long x )
```

Definition at line 91 of file literals.h.

5.4.1.730 operator""_MGy() [2/2]

```
quantity< absorbed_dose , long double> boost::units::literals::operator"" _MGy (   
    long double x )
```

Definition at line 91 of file literals.h.

5.4.1.731 operator""_mH() [1/2]

```
quantity< inductance , unsigned long long> boost::units::literals::operator"" _mH (   
    unsigned long long x )
```

Definition at line 86 of file literals.h.

5.4.1.732 operator""_MH() [1/2]

```
quantity< inductance , long double> boost::units::literals::operator"" _MH (   
    long double x )
```

Definition at line 86 of file literals.h.

5.4.1.733 operator""_MH() [2/2]

```
quantity< inductance , unsigned long long> boost::units::literals::operator""_MH (  
    unsigned long long x )
```

Definition at line 86 of file literals.h.

5.4.1.734 operator""_mH() [2/2]

```
quantity< inductance , long double> boost::units::literals::operator""_mH (   
    long double x )
```

Definition at line 86 of file literals.h.

5.4.1.735 operator""_Mh() [1/2]

```
quantity< time , long double> boost::units::literals::operator""_Mh (   
    long double x )
```

Definition at line 95 of file literals.h.

5.4.1.736 operator""_mh() [1/2]

```
quantity< time , long double> boost::units::literals::operator""_mh (   
    long double x )
```

Definition at line 95 of file literals.h.

5.4.1.737 operator""_mh() [2/2]

```
quantity< time , unsigned long long> boost::units::literals::operator""_mh (   
    unsigned long long x )
```

Definition at line 95 of file literals.h.

5.4.1.738 operator""_Mh() [2/2]

```
quantity< time , unsigned long long> boost::units::literals::operator""_Mh (   
    unsigned long long x )
```

Definition at line 95 of file literals.h.

5.4.1.739 operator""_MHz() [1/2]

```
quantity< frequency , long double> boost::units::literals::operator""_MHz (   
    long double x )
```

Definition at line 72 of file literals.h.

5.4.1.740 operator""_mHz() [1/2]

```
quantity< frequency , long double> boost::units::literals::operator"" _mHz ( long double x )
```

Definition at line 72 of file literals.h.

5.4.1.741 operator""_mHz() [2/2]

```
quantity< frequency , unsigned long long> boost::units::literals::operator"" _mHz ( unsigned long long x )
```

Definition at line 72 of file literals.h.

5.4.1.742 operator""_MHz() [2/2]

```
quantity< frequency , unsigned long long> boost::units::literals::operator"" _MHz ( unsigned long long x )
```

Definition at line 72 of file literals.h.

5.4.1.743 operator""_min() [1/2]

```
quantity< time , long double> boost::units::literals::operator"" _min ( long double x )
```

Definition at line 94 of file literals.h.

5.4.1.744 operator""_min() [2/2]

```
quantity< time , unsigned long long> boost::units::literals::operator"" _min ( unsigned long long x )
```

Definition at line 94 of file literals.h.

5.4.1.745 operator""_MJ() [1/2]

```
quantity< energy , long double> boost::units::literals::operator"" _MJ ( long double x )
```

Definition at line 77 of file literals.h.

5.4.1.746 operator""_MJ() [2/2]

```
quantity< energy , unsigned long long> boost::units::literals::operator"" _MJ ( unsigned long long x )
```

Definition at line 77 of file literals.h.

5.4.1.747 operator""_mJ() [1/2]

```
quantity< energy , long double> boost::units::literals::operator"" _mJ ( long double x )
```

Definition at line 77 of file literals.h.

5.4.1.748 operator""_mJ() [2/2]

```
quantity< energy , unsigned long long> boost::units::literals::operator"" _mJ ( unsigned long long x )
```

Definition at line 77 of file literals.h.

5.4.1.749 operator""_mK() [1/2]

```
quantity< temperature , long double> boost::units::literals::operator"" _mK ( long double x )
```

Definition at line 69 of file literals.h.

5.4.1.750 operator""_mK() [1/2]

```
quantity< temperature , long double> boost::units::literals::operator"" _mK ( long double x )
```

Definition at line 69 of file literals.h.

5.4.1.751 operator""_mK() [2/2]

```
quantity< temperature , unsigned long long> boost::units::literals::operator"" _mK ( unsigned long long x )
```

Definition at line 69 of file literals.h.

5.4.1.752 operator""_MK() [2/2]

```
quantity< temperature , unsigned long long> boost::units::literals::operator"" _MK ( unsigned long long x )
```

Definition at line 69 of file literals.h.

5.4.1.753 operator""_Mkat() [1/2]

```
quantity< catalytic_activity , long double> boost::units::literals::operator"" _Mkat ( long double x )
```

Definition at line 93 of file literals.h.

5.4.1.754 operator""_mkat() [1/2]

```
quantity< catalytic_activity , unsigned long long> boost::units::literals::operator"" _mkat (  
    unsigned long long x )
```

Definition at line 93 of file literals.h.

5.4.1.755 operator""_mkat() [2/2]

```
quantity< catalytic_activity , long double> boost::units::literals::operator"" _mkat (   
    long double x )
```

Definition at line 93 of file literals.h.

5.4.1.756 operator""_Mkat() [2/2]

```
quantity< catalytic_activity , unsigned long long> boost::units::literals::operator"" _Mkat (   
    unsigned long long x )
```

Definition at line 93 of file literals.h.

5.4.1.757 operator""_Ml() [1/2]

```
quantity< volume , unsigned long long> boost::units::literals::operator"" _Ml (   
    unsigned long long x )
```

Definition at line 98 of file literals.h.

5.4.1.758 operator""_ml() [1/2]

```
quantity< volume , long double> boost::units::literals::operator"" _ml (   
    long double x )
```

Definition at line 98 of file literals.h.

5.4.1.759 operator""_ml() [2/2]

```
quantity< volume , unsigned long long> boost::units::literals::operator"" _ml (   
    unsigned long long x )
```

Definition at line 98 of file literals.h.

5.4.1.760 operator""_Ml() [2/2]

```
quantity< volume , long double> boost::units::literals::operator"" _Ml (   
    long double x )
```

Definition at line 98 of file literals.h.

5.4.1.761 operator""_ML() [1/2]

```
quantity< volume , unsigned long long> boost::units::literals::operator"" _ML (  
    unsigned long long x )
```

Definition at line 99 of file literals.h.

5.4.1.762 operator""_mL() [1/2]

```
quantity< volume , long double> boost::units::literals::operator"" _mL (   
    long double x )
```

Definition at line 99 of file literals.h.

5.4.1.763 operator""_mL() [2/2]

```
quantity< volume , unsigned long long> boost::units::literals::operator"" _mL (   
    unsigned long long x )
```

Definition at line 99 of file literals.h.

5.4.1.764 operator""_ML() [2/2]

```
quantity< volume , long double> boost::units::literals::operator"" _ML (   
    long double x )
```

Definition at line 99 of file literals.h.

5.4.1.765 operator""_Mlm() [1/2]

```
quantity< luminous_flux , long double> boost::units::literals::operator"" _Mlm (   
    long double x )
```

Definition at line 88 of file literals.h.

5.4.1.766 operator""_Mlm() [2/2]

```
quantity< luminous_flux , unsigned long long> boost::units::literals::operator"" _Mlm (   
    unsigned long long x )
```

Definition at line 88 of file literals.h.

5.4.1.767 operator""_mlm() [1/2]

```
quantity< luminous_flux , long double> boost::units::literals::operator"" _mlm (   
    long double x )
```

Definition at line 88 of file literals.h.

5.4.1.768 operator""_mlm() [2/2]

```
quantity< luminous_flux , unsigned long long> boost::units::literals::operator""_mlm (  
    unsigned long long x )
```

Definition at line 88 of file literals.h.

5.4.1.769 operator""_mlx() [1/2]

```
quantity< illuminance , long double> boost::units::literals::operator""_mlx (   
    long double x )
```

Definition at line 89 of file literals.h.

5.4.1.770 operator""_mlx() [2/2]

```
quantity< illuminance , unsigned long long> boost::units::literals::operator""_mlx (   
    unsigned long long x )
```

Definition at line 89 of file literals.h.

5.4.1.771 operator""_Mlx() [1/2]

```
quantity< illuminance , long double> boost::units::literals::operator""_Mlx (   
    long double x )
```

Definition at line 89 of file literals.h.

5.4.1.772 operator""_Mlx() [2/2]

```
quantity< illuminance , unsigned long long> boost::units::literals::operator""_Mlx (   
    unsigned long long x )
```

Definition at line 89 of file literals.h.

5.4.1.773 operator""_Mm() [1/2]

```
quantity< length , unsigned long long> boost::units::literals::operator""_Mm (   
    unsigned long long x )
```

Definition at line 65 of file literals.h.

5.4.1.774 operator""_Mm() [2/2]

```
quantity< length , long double> boost::units::literals::operator""_Mm (   
    long double x )
```

Definition at line 65 of file literals.h.

5.4.1.775 operator""_mm() [1/2]

```
quantity< length , long double> boost::units::literals::operator"" _mm ( long double x )
```

Definition at line 65 of file literals.h.

5.4.1.776 operator""_mm() [2/2]

```
quantity< length , unsigned long long> boost::units::literals::operator"" _mm ( unsigned long long x )
```

Definition at line 65 of file literals.h.

5.4.1.777 operator""_mmin() [1/2]

```
quantity< time , long double> boost::units::literals::operator"" _mmin ( long double x )
```

Definition at line 94 of file literals.h.

5.4.1.778 operator""_Mmin() [1/2]

```
quantity< time , long double> boost::units::literals::operator"" _Mmin ( long double x )
```

Definition at line 94 of file literals.h.

5.4.1.779 operator""_Mmin() [2/2]

```
quantity< time , unsigned long long> boost::units::literals::operator"" _Mmin ( unsigned long long x )
```

Definition at line 94 of file literals.h.

5.4.1.780 operator""_mmin() [2/2]

```
quantity< time , unsigned long long> boost::units::literals::operator"" _mmin ( unsigned long long x )
```

Definition at line 94 of file literals.h.

5.4.1.781 operator""_Mmol() [1/2]

```
quantity< amount , long double> boost::units::literals::operator"" _Mmol ( long double x )
```

Definition at line 70 of file literals.h.

5.4.1.782 operator""_Mmol() [2/2]

```
quantity< amount , unsigned long long> boost::units::literals::operator"" _Mmol (  
    unsigned long long x )
```

Definition at line 70 of file literals.h.

5.4.1.783 operator""_mmol() [1/2]

```
quantity< amount , long double> boost::units::literals::operator"" _mmol (   
    long double x )
```

Definition at line 70 of file literals.h.

5.4.1.784 operator""_mmol() [2/2]

```
quantity< amount , unsigned long long> boost::units::literals::operator"" _mmol (   
    unsigned long long x )
```

Definition at line 70 of file literals.h.

5.4.1.785 operator""_MN() [1/2]

```
quantity< force , long double> boost::units::literals::operator"" _MN (   
    long double x )
```

Definition at line 75 of file literals.h.

5.4.1.786 operator""_MN() [2/2]

```
quantity< force , unsigned long long> boost::units::literals::operator"" _MN (   
    unsigned long long x )
```

Definition at line 75 of file literals.h.

5.4.1.787 operator""_mN() [1/2]

```
quantity< force , long double> boost::units::literals::operator"" _mN (   
    long double x )
```

Definition at line 75 of file literals.h.

5.4.1.788 operator""_mN() [2/2]

```
quantity< force , unsigned long long> boost::units::literals::operator"" _mN (   
    unsigned long long x )
```

Definition at line 75 of file literals.h.

5.4.1.789 operator""_Mohm() [1/2]

```
quantity< resistance , unsigned long long> boost::units::literals::operator""_Mohm (  
    unsigned long long x )
```

Definition at line 82 of file literals.h.

5.4.1.790 operator""_mohm() [1/2]

```
quantity< resistance , unsigned long long> boost::units::literals::operator""_mohm (   
    unsigned long long x )
```

Definition at line 82 of file literals.h.

5.4.1.791 operator""_Mohm() [2/2]

```
quantity< resistance , long double> boost::units::literals::operator""_Mohm (   
    long double x )
```

Definition at line 82 of file literals.h.

5.4.1.792 operator""_mohm() [2/2]

```
quantity< resistance , long double> boost::units::literals::operator""_mohm (   
    long double x )
```

Definition at line 82 of file literals.h.

5.4.1.793 operator""_mol() [1/2]

```
quantity< amount , long double> boost::units::literals::operator""_mol (   
    long double x )
```

Definition at line 70 of file literals.h.

5.4.1.794 operator""_mol() [2/2]

```
quantity< amount , unsigned long long> boost::units::literals::operator""_mol (   
    unsigned long long x )
```

Definition at line 70 of file literals.h.

5.4.1.795 operator""_mPa() [1/2]

```
quantity< pressure , unsigned long long> boost::units::literals::operator""_mPa (   
    unsigned long long x )
```

Definition at line 76 of file literals.h.

5.4.1.796 operator""_MPa() [1/2]

```
quantity< pressure , unsigned long long> boost::units::literals::operator"" _MPa (  
    unsigned long long x )
```

Definition at line 76 of file literals.h.

5.4.1.797 operator""_MPa() [2/2]

```
quantity< pressure , long double> boost::units::literals::operator"" _MPa (   
    long double x )
```

Definition at line 76 of file literals.h.

5.4.1.798 operator""_mPa() [2/2]

```
quantity< pressure , long double> boost::units::literals::operator"" _mPa (   
    long double x )
```

Definition at line 76 of file literals.h.

5.4.1.799 operator""_Mrad() [1/2]

```
quantity< plane_angle , long double> boost::units::literals::operator"" _Mrad (   
    long double x )
```

Definition at line 73 of file literals.h.

5.4.1.800 operator""_mrad() [1/2]

```
quantity< plane_angle , unsigned long long> boost::units::literals::operator"" _mrad (   
    unsigned long long x )
```

Definition at line 73 of file literals.h.

5.4.1.801 operator""_mrad() [2/2]

```
quantity< plane_angle , long double> boost::units::literals::operator"" _mrad (   
    long double x )
```

Definition at line 73 of file literals.h.

5.4.1.802 operator""_Mrad() [2/2]

```
quantity< plane_angle , unsigned long long> boost::units::literals::operator"" _Mrad (   
    unsigned long long x )
```

Definition at line 73 of file literals.h.

5.4.1.803 operator""_Ms() [1/2]

```
quantity< time , long double> boost::units::literals::operator"" _Ms ( long double x )
```

Definition at line 67 of file literals.h.

5.4.1.804 operator""_Ms() [2/2]

```
quantity< time , unsigned long long> boost::units::literals::operator"" _Ms ( unsigned long long x )
```

Definition at line 67 of file literals.h.

5.4.1.805 operator""_ms() [1/2]

```
quantity< time , long double> boost::units::literals::operator"" _ms ( long double x )
```

Definition at line 67 of file literals.h.

5.4.1.806 operator""_ms() [2/2]

```
quantity< time , unsigned long long> boost::units::literals::operator"" _ms ( unsigned long long x )
```

Definition at line 67 of file literals.h.

5.4.1.807 operator""_MS() [1/2]

```
quantity< conductance , long double> boost::units::literals::operator"" _MS ( long double x )
```

Definition at line 83 of file literals.h.

5.4.1.808 operator""_MS() [2/2]

```
quantity< conductance , unsigned long long> boost::units::literals::operator"" _MS ( unsigned long long x )
```

Definition at line 83 of file literals.h.

5.4.1.809 operator""_mS() [1/2]

```
quantity< conductance , long double> boost::units::literals::operator"" _mS ( long double x )
```

Definition at line 83 of file literals.h.

5.4.1.810 operator""_mS() [2/2]

```
quantity< conductance , unsigned long long> boost::units::literals::operator"" _mS ( unsigned long long x )
```

Definition at line 83 of file literals.h.

5.4.1.811 operator""_Msr() [1/2]

```
quantity< solid_angle , long double> boost::units::literals::operator"" _Msr ( long double x )
```

Definition at line 74 of file literals.h.

5.4.1.812 operator""_Msr() [2/2]

```
quantity< solid_angle , unsigned long long> boost::units::literals::operator"" _Msr ( unsigned long long x )
```

Definition at line 74 of file literals.h.

5.4.1.813 operator""_msr() [1/2]

```
quantity< solid_angle , long double> boost::units::literals::operator"" _msr ( long double x )
```

Definition at line 74 of file literals.h.

5.4.1.814 operator""_msr() [2/2]

```
quantity< solid_angle , unsigned long long> boost::units::literals::operator"" _msr ( unsigned long long x )
```

Definition at line 74 of file literals.h.

5.4.1.815 operator""_mSv() [1/2]

```
quantity< dose_equivalent , long double> boost::units::literals::operator"" _mSv ( long double x )
```

Definition at line 92 of file literals.h.

5.4.1.816 operator""_mSv() [2/2]

```
quantity< dose_equivalent , unsigned long long> boost::units::literals::operator"" _mSv ( unsigned long long x )
```

Definition at line 92 of file literals.h.

5.4.1.817 operator""_MSv() [1/2]

```
quantity< dose_equivalent , long double> boost::units::literals::operator""_MSv ( long double x )
```

Definition at line 92 of file literals.h.

5.4.1.818 operator""_MSv() [2/2]

```
quantity< dose_equivalent , unsigned long long> boost::units::literals::operator""_MSv ( unsigned long long x )
```

Definition at line 92 of file literals.h.

5.4.1.819 operator""_MT() [1/2]

```
quantity< magnetic_flux_density , long double> boost::units::literals::operator""_MT ( long double x )
```

Definition at line 85 of file literals.h.

5.4.1.820 operator""_MT() [2/2]

```
quantity< magnetic_flux_density , unsigned long long> boost::units::literals::operator""_MT ( unsigned long long x )
```

Definition at line 85 of file literals.h.

5.4.1.821 operator""_mT() [1/2]

```
quantity< magnetic_flux_density , long double> boost::units::literals::operator""_mT ( long double x )
```

Definition at line 85 of file literals.h.

5.4.1.822 operator""_mT() [2/2]

```
quantity< magnetic_flux_density , unsigned long long> boost::units::literals::operator""_mT ( unsigned long long x )
```

Definition at line 85 of file literals.h.

5.4.1.823 operator""_Mt() [1/2]

```
quantity< mass , unsigned long long> boost::units::literals::operator""_Mt ( unsigned long long x )
```

Definition at line 100 of file literals.h.

5.4.1.824 operator""_mt() [1/2]

```
quantity< mass , long double> boost::units::literals::operator""_mt (  
    long double x )
```

Definition at line 100 of file literals.h.

5.4.1.825 operator""_Mt() [2/2]

```
quantity< mass , long double> boost::units::literals::operator""_Mt (   
    long double x )
```

Definition at line 100 of file literals.h.

5.4.1.826 operator""_mt() [2/2]

```
quantity< mass , unsigned long long> boost::units::literals::operator""_mt (   
    unsigned long long x )
```

Definition at line 100 of file literals.h.

5.4.1.827 operator""_MV() [1/2]

```
quantity< electric_potential , long double> boost::units::literals::operator""_MV (   
    long double x )
```

Definition at line 80 of file literals.h.

5.4.1.828 operator""_MV() [2/2]

```
quantity< electric_potential , unsigned long long> boost::units::literals::operator""_MV (   
    unsigned long long x )
```

Definition at line 80 of file literals.h.

5.4.1.829 operator""_mV() [1/2]

```
quantity< electric_potential , long double> boost::units::literals::operator""_mV (   
    long double x )
```

Definition at line 80 of file literals.h.

5.4.1.830 operator""_mV() [2/2]

```
quantity< electric_potential , unsigned long long> boost::units::literals::operator""_mV (   
    unsigned long long x )
```

Definition at line 80 of file literals.h.

5.4.1.831 operator""_MW() [1/2]

```
quantity< power , unsigned long long> boost::units::literals::operator"" _MW ( unsigned long long x )
```

Definition at line 78 of file literals.h.

5.4.1.832 operator""_mW() [1/2]

```
quantity< power , long double> boost::units::literals::operator"" _mW ( long double x )
```

Definition at line 78 of file literals.h.

5.4.1.833 operator""_MW() [2/2]

```
quantity< power , long double> boost::units::literals::operator"" _MW ( long double x )
```

Definition at line 78 of file literals.h.

5.4.1.834 operator""_mW() [2/2]

```
quantity< power , unsigned long long> boost::units::literals::operator"" _mW ( unsigned long long x )
```

Definition at line 78 of file literals.h.

5.4.1.835 operator""_MWb() [1/2]

```
quantity< magnetic_flux , long double> boost::units::literals::operator"" _MWb ( long double x )
```

Definition at line 84 of file literals.h.

5.4.1.836 operator""_MWb() [2/2]

```
quantity< magnetic_flux , unsigned long long> boost::units::literals::operator"" _MWb ( unsigned long long x )
```

Definition at line 84 of file literals.h.

5.4.1.837 operator""_mWb() [1/2]

```
quantity< magnetic_flux , long double> boost::units::literals::operator"" _mWb ( long double x )
```

Definition at line 84 of file literals.h.

5.4.1.838 operator""_mWb() [2/2]

```
quantity< magnetic_flux , unsigned long long> boost::units::literals::operator""_mWb ( unsigned long long x )
```

Definition at line 84 of file literals.h.

5.4.1.839 operator""_N() [1/2]

```
quantity< force , long double> boost::units::literals::operator""_N ( long double x )
```

Definition at line 75 of file literals.h.

5.4.1.840 operator""_N() [2/2]

```
quantity< force , unsigned long long> boost::units::literals::operator""_N ( unsigned long long x )
```

Definition at line 75 of file literals.h.

5.4.1.841 operator""_nA() [1/2]

```
quantity< current , long double> boost::units::literals::operator""_nA ( long double x )
```

Definition at line 68 of file literals.h.

5.4.1.842 operator""_nA() [2/2]

```
quantity< current , unsigned long long> boost::units::literals::operator""_nA ( unsigned long long x )
```

Definition at line 68 of file literals.h.

5.4.1.843 operator""_nBq() [1/2]

```
quantity< activity , long double> boost::units::literals::operator""_nBq ( long double x )
```

Definition at line 90 of file literals.h.

5.4.1.844 operator""_nBq() [2/2]

```
quantity< activity , unsigned long long> boost::units::literals::operator""_nBq ( unsigned long long x )
```

Definition at line 90 of file literals.h.

5.4.1.845 operator""_nC() [1/2]

```
quantity< electric_charge , unsigned long long> boost::units::literals::operator"" _nC (
    unsigned long long x )
```

Definition at line 79 of file literals.h.

5.4.1.846 operator""_nC() [2/2]

```
quantity< electric_charge , long double> boost::units::literals::operator"" _nC (
    long double x )
```

Definition at line 79 of file literals.h.

5.4.1.847 operator""_ncd() [1/2]

```
quantity< luminous_intensity , long double> boost::units::literals::operator"" _ncd (
    long double x )
```

Definition at line 71 of file literals.h.

5.4.1.848 operator""_ncd() [2/2]

```
quantity< luminous_intensity , unsigned long long> boost::units::literals::operator"" _ncd (
    unsigned long long x )
```

Definition at line 71 of file literals.h.

5.4.1.849 operator""_nday() [1/2]

```
quantity< time , long double> boost::units::literals::operator"" _nday (
    long double x )
```

Definition at line 96 of file literals.h.

5.4.1.850 operator""_nday() [2/2]

```
quantity< time , unsigned long long> boost::units::literals::operator"" _nday (
    unsigned long long x )
```

Definition at line 96 of file literals.h.

5.4.1.851 operator""_ndeg() [1/2]

```
quantity< plane_angle , unsigned long long> boost::units::literals::operator"" _ndeg (
    unsigned long long x )
```

Definition at line 97 of file literals.h.

5.4.1.852 operator""_ndeg() [2/2]

```
quantity< plane_angle , long double> boost::units::literals::operator""_ndeg ( long double x )
```

Definition at line 97 of file literals.h.

5.4.1.853 operator""_ndegC() [1/2]

```
quantity< temperature , long double> boost::units::literals::operator""_ndegC ( long double x )
```

Definition at line 87 of file literals.h.

5.4.1.854 operator""_ndegC() [2/2]

```
quantity< temperature , unsigned long long> boost::units::literals::operator""_ndegC ( unsigned long long x )
```

Definition at line 87 of file literals.h.

5.4.1.855 operator""_nF() [1/2]

```
quantity< capacitance , long double> boost::units::literals::operator""_nF ( long double x )
```

Definition at line 81 of file literals.h.

5.4.1.856 operator""_nF() [2/2]

```
quantity< capacitance , unsigned long long> boost::units::literals::operator""_nF ( unsigned long long x )
```

Definition at line 81 of file literals.h.

5.4.1.857 operator""_ng() [1/2]

```
quantity< mass , unsigned long long> boost::units::literals::operator""_ng ( unsigned long long x )
```

Definition at line 66 of file literals.h.

5.4.1.858 operator""_ng() [2/2]

```
quantity< mass , long double> boost::units::literals::operator""_ng ( long double x )
```

Definition at line 66 of file literals.h.

5.4.1.859 operator""_nGy() [1/2]

```
quantity< absorbed_dose , unsigned long long> boost::units::literals::operator"" _nGy (
    unsigned long long x )
```

Definition at line 91 of file literals.h.

5.4.1.860 operator""_nGy() [2/2]

```
quantity< absorbed_dose , long double> boost::units::literals::operator"" _nGy (
    long double x )
```

Definition at line 91 of file literals.h.

5.4.1.861 operator""_nH() [1/2]

```
quantity< inductance , long double> boost::units::literals::operator"" _nH (
    long double x )
```

Definition at line 86 of file literals.h.

5.4.1.862 operator""_nH() [2/2]

```
quantity< inductance , unsigned long long> boost::units::literals::operator"" _nH (
    unsigned long long x )
```

Definition at line 86 of file literals.h.

5.4.1.863 operator""_nh() [1/2]

```
quantity< time , long double> boost::units::literals::operator"" _nh (
    long double x )
```

Definition at line 95 of file literals.h.

5.4.1.864 operator""_nh() [2/2]

```
quantity< time , unsigned long long> boost::units::literals::operator"" _nh (
    unsigned long long x )
```

Definition at line 95 of file literals.h.

5.4.1.865 operator""_nHz() [1/2]

```
quantity< frequency , long double> boost::units::literals::operator"" _nHz (
    long double x )
```

Definition at line 72 of file literals.h.

5.4.1.866 operator""_nHz() [2/2]

```
quantity< frequency , unsigned long long> boost::units::literals::operator""_nHz (  
    unsigned long long x )
```

Definition at line 72 of file literals.h.

5.4.1.867 operator""_nJ() [1/2]

```
quantity< energy , unsigned long long> boost::units::literals::operator""_nJ (   
    unsigned long long x )
```

Definition at line 77 of file literals.h.

5.4.1.868 operator""_nJ() [2/2]

```
quantity< energy , long double> boost::units::literals::operator""_nJ (   
    long double x )
```

Definition at line 77 of file literals.h.

5.4.1.869 operator""_nK() [1/2]

```
quantity< temperature , long double> boost::units::literals::operator""_nK (   
    long double x )
```

Definition at line 69 of file literals.h.

5.4.1.870 operator""_nK() [2/2]

```
quantity< temperature , unsigned long long> boost::units::literals::operator""_nK (   
    unsigned long long x )
```

Definition at line 69 of file literals.h.

5.4.1.871 operator""_nkat() [1/2]

```
quantity< catalytic_activity , unsigned long long> boost::units::literals::operator""_nkat (   
    unsigned long long x )
```

Definition at line 93 of file literals.h.

5.4.1.872 operator""_nkat() [2/2]

```
quantity< catalytic_activity , long double> boost::units::literals::operator""_nkat (   
    long double x )
```

Definition at line 93 of file literals.h.

5.4.1.873 operator""_nl() [1/2]

```
quantity< volume , long double> boost::units::literals::operator""_nl (  
    long double x )
```

Definition at line 98 of file literals.h.

5.4.1.874 operator""_nl() [2/2]

```
quantity< volume , unsigned long long> boost::units::literals::operator""_nl (  
    unsigned long long x )
```

Definition at line 98 of file literals.h.

5.4.1.875 operator""_nL() [1/2]

```
quantity< volume , unsigned long long> boost::units::literals::operator""_nL (   
    unsigned long long x )
```

Definition at line 99 of file literals.h.

5.4.1.876 operator""_nL() [2/2]

```
quantity< volume , long double> boost::units::literals::operator""_nL (   
    long double x )
```

Definition at line 99 of file literals.h.

5.4.1.877 operator""_nlm() [1/2]

```
quantity< luminous_flux , long double> boost::units::literals::operator""_nlm (   
    long double x )
```

Definition at line 88 of file literals.h.

5.4.1.878 operator""_nlm() [2/2]

```
quantity< luminous_flux , unsigned long long> boost::units::literals::operator""_nlm (   
    unsigned long long x )
```

Definition at line 88 of file literals.h.

5.4.1.879 operator""_nlx() [1/2]

```
quantity< illuminance , long double> boost::units::literals::operator""_nlx (   
    long double x )
```

Definition at line 89 of file literals.h.

5.4.1.880 operator""_nlx() [2/2]

```
quantity< illuminance , unsigned long long> boost::units::literals::operator""_nlx (  
    unsigned long long x )
```

Definition at line 89 of file literals.h.

5.4.1.881 operator""_nm() [1/2]

```
quantity< length , unsigned long long> boost::units::literals::operator""_nm (   
    unsigned long long x )
```

Definition at line 65 of file literals.h.

5.4.1.882 operator""_nm() [2/2]

```
quantity< length , long double> boost::units::literals::operator""_nm (   
    long double x )
```

Definition at line 65 of file literals.h.

5.4.1.883 operator""_nmin() [1/2]

```
quantity< time , long double> boost::units::literals::operator""_nmin (   
    long double x )
```

Definition at line 94 of file literals.h.

5.4.1.884 operator""_nmin() [2/2]

```
quantity< time , unsigned long long> boost::units::literals::operator""_nmin (   
    unsigned long long x )
```

Definition at line 94 of file literals.h.

5.4.1.885 operator""_nmol() [1/2]

```
quantity< amount , long double> boost::units::literals::operator""_nmol (   
    long double x )
```

Definition at line 70 of file literals.h.

5.4.1.886 operator""_nmol() [2/2]

```
quantity< amount , unsigned long long> boost::units::literals::operator""_nmol (   
    unsigned long long x )
```

Definition at line 70 of file literals.h.

5.4.1.887 operator""_nN() [1/2]

```
quantity< force , long double> boost::units::literals::operator""_nN ( long double x )
```

Definition at line 75 of file literals.h.

5.4.1.888 operator""_nN() [2/2]

```
quantity< force , unsigned long long> boost::units::literals::operator""_nN ( unsigned long long x )
```

Definition at line 75 of file literals.h.

5.4.1.889 operator""_nohm() [1/2]

```
quantity< resistance , unsigned long long> boost::units::literals::operator""_nohm ( unsigned long long x )
```

Definition at line 82 of file literals.h.

5.4.1.890 operator""_nohm() [2/2]

```
quantity< resistance , long double> boost::units::literals::operator""_nohm ( long double x )
```

Definition at line 82 of file literals.h.

5.4.1.891 operator""_nPa() [1/2]

```
quantity< pressure , unsigned long long> boost::units::literals::operator""_nPa ( unsigned long long x )
```

Definition at line 76 of file literals.h.

5.4.1.892 operator""_nPa() [2/2]

```
quantity< pressure , long double> boost::units::literals::operator""_nPa ( long double x )
```

Definition at line 76 of file literals.h.

5.4.1.893 operator""_nrad() [1/2]

```
quantity< plane_angle , long double> boost::units::literals::operator""_nrad ( long double x )
```

Definition at line 73 of file literals.h.

5.4.1.894 operator""_nrad() [2/2]

```
quantity< plane_angle , unsigned long long> boost::units::literals::operator"" _nrad (  
    unsigned long long x )
```

Definition at line 73 of file literals.h.

5.4.1.895 operator""_ns() [1/2]

```
quantity< time , long double> boost::units::literals::operator"" _ns (   
    long double x )
```

Definition at line 67 of file literals.h.

5.4.1.896 operator""_ns() [2/2]

```
quantity< time , unsigned long long> boost::units::literals::operator"" _ns (   
    unsigned long long x )
```

Definition at line 67 of file literals.h.

5.4.1.897 operator""_nS() [1/2]

```
quantity< conductance , long double> boost::units::literals::operator"" _nS (   
    long double x )
```

Definition at line 83 of file literals.h.

5.4.1.898 operator""_nS() [2/2]

```
quantity< conductance , unsigned long long> boost::units::literals::operator"" _nS (   
    unsigned long long x )
```

Definition at line 83 of file literals.h.

5.4.1.899 operator""_nsr() [1/2]

```
quantity< solid_angle , long double> boost::units::literals::operator"" _nsr (   
    long double x )
```

Definition at line 74 of file literals.h.

5.4.1.900 operator""_nsr() [2/2]

```
quantity< solid_angle , unsigned long long> boost::units::literals::operator"" _nsr (   
    unsigned long long x )
```

Definition at line 74 of file literals.h.

5.4.1.901 operator""_nSv() [1/2]

```
quantity< dose_equivalent , long double> boost::units::literals::operator"" _nSv ( long double x )
```

Definition at line 92 of file literals.h.

5.4.1.902 operator""_nSv() [2/2]

```
quantity< dose_equivalent , unsigned long long> boost::units::literals::operator"" _nSv ( unsigned long long x )
```

Definition at line 92 of file literals.h.

5.4.1.903 operator""_nT() [1/2]

```
quantity< magnetic_flux_density , long double> boost::units::literals::operator"" _nT ( long double x )
```

Definition at line 85 of file literals.h.

5.4.1.904 operator""_nT() [2/2]

```
quantity< magnetic_flux_density , unsigned long long> boost::units::literals::operator"" _nT ( unsigned long long x )
```

Definition at line 85 of file literals.h.

5.4.1.905 operator""_nt() [1/2]

```
quantity< mass , long double> boost::units::literals::operator"" _nt ( long double x )
```

Definition at line 100 of file literals.h.

5.4.1.906 operator""_nt() [2/2]

```
quantity< mass , unsigned long long> boost::units::literals::operator"" _nt ( unsigned long long x )
```

Definition at line 100 of file literals.h.

5.4.1.907 operator""_nV() [1/2]

```
quantity< electric_potential , unsigned long long> boost::units::literals::operator"" _nV ( unsigned long long x )
```

Definition at line 80 of file literals.h.

5.4.1.908 operator""_nV() [2/2]

```
quantity< electric_potential , long double> boost::units::literals::operator"" _nV ( long double x )
```

Definition at line 80 of file literals.h.

5.4.1.909 operator""_nW() [1/2]

```
quantity< power , long double> boost::units::literals::operator"" _nW ( long double x )
```

Definition at line 78 of file literals.h.

5.4.1.910 operator""_nW() [2/2]

```
quantity< power , unsigned long long> boost::units::literals::operator"" _nW ( unsigned long long x )
```

Definition at line 78 of file literals.h.

5.4.1.911 operator""_nWb() [1/2]

```
quantity< magnetic_flux , long double> boost::units::literals::operator"" _nWb ( long double x )
```

Definition at line 84 of file literals.h.

5.4.1.912 operator""_nWb() [2/2]

```
quantity< magnetic_flux , unsigned long long> boost::units::literals::operator"" _nWb ( unsigned long long x )
```

Definition at line 84 of file literals.h.

5.4.1.913 operator""_ohm() [1/2]

```
quantity< resistance , long double> boost::units::literals::operator"" _ohm ( long double x )
```

Definition at line 82 of file literals.h.

5.4.1.914 operator""_ohm() [2/2]

```
quantity< resistance , unsigned long long> boost::units::literals::operator"" _ohm ( unsigned long long x )
```

Definition at line 82 of file literals.h.

5.4.1.915 operator""_pA() [1/2]

```
quantity< current , unsigned long long> boost::units::literals::operator"" _pA (  
    unsigned long long x )
```

Definition at line 68 of file literals.h.

5.4.1.916 operator""_pA() [2/2]

```
quantity< current , long double> boost::units::literals::operator"" _pA (   
    long double x )
```

Definition at line 68 of file literals.h.

5.4.1.917 operator""_PA() [1/2]

```
quantity< current , unsigned long long> boost::units::literals::operator"" _PA (   
    unsigned long long x )
```

Definition at line 68 of file literals.h.

5.4.1.918 operator""_PA() [2/2]

```
quantity< current , long double> boost::units::literals::operator"" _PA (   
    long double x )
```

Definition at line 68 of file literals.h.

5.4.1.919 operator""_Pa() [1/2]

```
quantity< pressure , long double> boost::units::literals::operator"" _Pa (   
    long double x )
```

Definition at line 76 of file literals.h.

5.4.1.920 operator""_Pa() [2/2]

```
quantity< pressure , unsigned long long> boost::units::literals::operator"" _Pa (   
    unsigned long long x )
```

Definition at line 76 of file literals.h.

5.4.1.921 operator""_PBq() [1/2]

```
quantity< activity , long double> boost::units::literals::operator"" _PBq (   
    long double x )
```

Definition at line 90 of file literals.h.

5.4.1.922 operator""_PBq() [2/2]

```
quantity< activity , unsigned long long> boost::units::literals::operator"" _PBq (  
    unsigned long long x )
```

Definition at line 90 of file literals.h.

5.4.1.923 operator""_pBq() [1/2]

```
quantity< activity , unsigned long long> boost::units::literals::operator"" _pBq (   
    unsigned long long x )
```

Definition at line 90 of file literals.h.

5.4.1.924 operator""_pBq() [2/2]

```
quantity< activity , long double> boost::units::literals::operator"" _pBq (   
    long double x )
```

Definition at line 90 of file literals.h.

5.4.1.925 operator""_PC() [1/2]

```
quantity< electric_charge , unsigned long long> boost::units::literals::operator"" _PC (   
    unsigned long long x )
```

Definition at line 79 of file literals.h.

5.4.1.926 operator""_pC() [1/2]

```
quantity< electric_charge , long double> boost::units::literals::operator"" _pC (   
    long double x )
```

Definition at line 79 of file literals.h.

5.4.1.927 operator""_pC() [2/2]

```
quantity< electric_charge , unsigned long long> boost::units::literals::operator"" _pC (   
    unsigned long long x )
```

Definition at line 79 of file literals.h.

5.4.1.928 operator""_PC() [2/2]

```
quantity< electric_charge , long double> boost::units::literals::operator"" _PC (   
    long double x )
```

Definition at line 79 of file literals.h.

5.4.1.929 operator""_Pcd() [1/2]

```
quantity< luminous_intensity , unsigned long long> boost::units::literals::operator"" _Pcd (  
    unsigned long long x )
```

Definition at line 71 of file literals.h.

5.4.1.930 operator""_pcd() [1/2]

```
quantity< luminous_intensity , long double> boost::units::literals::operator"" _pcd (   
    long double x )
```

Definition at line 71 of file literals.h.

5.4.1.931 operator""_pcd() [2/2]

```
quantity< luminous_intensity , unsigned long long> boost::units::literals::operator"" _pcd (   
    unsigned long long x )
```

Definition at line 71 of file literals.h.

5.4.1.932 operator""_Pcd() [2/2]

```
quantity< luminous_intensity , long double> boost::units::literals::operator"" _Pcd (   
    long double x )
```

Definition at line 71 of file literals.h.

5.4.1.933 operator""_pday() [1/2]

```
quantity< time , long double> boost::units::literals::operator"" _pday (   
    long double x )
```

Definition at line 96 of file literals.h.

5.4.1.934 operator""_Pday() [1/2]

```
quantity< time , long double> boost::units::literals::operator"" _Pday (   
    long double x )
```

Definition at line 96 of file literals.h.

5.4.1.935 operator""_Pday() [2/2]

```
quantity< time , unsigned long long> boost::units::literals::operator"" _Pday (   
    unsigned long long x )
```

Definition at line 96 of file literals.h.

5.4.1.936 operator""_pday() [2/2]

```
quantity< time , unsigned long long> boost::units::literals::operator"" _pday (  
    unsigned long long x )
```

Definition at line 96 of file literals.h.

5.4.1.937 operator""_Pdeg() [1/2]

```
quantity< plane_angle , long double> boost::units::literals::operator"" _Pdeg (   
    long double x )
```

Definition at line 97 of file literals.h.

5.4.1.938 operator""_Pdeg() [2/2]

```
quantity< plane_angle , unsigned long long> boost::units::literals::operator"" _Pdeg (   
    unsigned long long x )
```

Definition at line 97 of file literals.h.

5.4.1.939 operator""_pdeg() [1/2]

```
quantity< plane_angle , unsigned long long> boost::units::literals::operator"" _pdeg (   
    unsigned long long x )
```

Definition at line 97 of file literals.h.

5.4.1.940 operator""_pdeg() [2/2]

```
quantity< plane_angle , long double> boost::units::literals::operator"" _pdeg (   
    long double x )
```

Definition at line 97 of file literals.h.

5.4.1.941 operator""_pdegC() [1/2]

```
quantity< temperature , long double> boost::units::literals::operator"" _pdegC (   
    long double x )
```

Definition at line 87 of file literals.h.

5.4.1.942 operator""_pdegC() [2/2]

```
quantity< temperature , unsigned long long> boost::units::literals::operator"" _pdegC (   
    unsigned long long x )
```

Definition at line 87 of file literals.h.

5.4.1.943 operator""_PdegC() [1/2]

```
quantity< temperature , long double> boost::units::literals::operator"" _PdegC (
    long double x )
```

Definition at line 87 of file literals.h.

5.4.1.944 operator""_PdegC() [2/2]

```
quantity< temperature , unsigned long long> boost::units::literals::operator"" _PdegC (
    unsigned long long x )
```

Definition at line 87 of file literals.h.

5.4.1.945 operator""_pF() [1/2]

```
quantity< capacitance , long double> boost::units::literals::operator"" _pF (
    long double x )
```

Definition at line 81 of file literals.h.

5.4.1.946 operator""_PF() [1/2]

```
quantity< capacitance , unsigned long long> boost::units::literals::operator"" _PF (
    unsigned long long x )
```

Definition at line 81 of file literals.h.

5.4.1.947 operator""_PF() [2/2]

```
quantity< capacitance , long double> boost::units::literals::operator"" _PF (
    long double x )
```

Definition at line 81 of file literals.h.

5.4.1.948 operator""_pF() [2/2]

```
quantity< capacitance , unsigned long long> boost::units::literals::operator"" _pF (
    unsigned long long x )
```

Definition at line 81 of file literals.h.

5.4.1.949 operator""_pg() [1/2]

```
quantity< mass , long double> boost::units::literals::operator"" _pg (
    long double x )
```

Definition at line 66 of file literals.h.

5.4.1.950 operator""_pg() [2/2]

```
quantity< mass , unsigned long long> boost::units::literals::operator""_pg (  
    unsigned long long x )
```

Definition at line 66 of file literals.h.

5.4.1.951 operator""_Pg() [1/2]

```
quantity< mass , long double> boost::units::literals::operator""_Pg (   
    long double x )
```

Definition at line 66 of file literals.h.

5.4.1.952 operator""_Pg() [2/2]

```
quantity< mass , unsigned long long> boost::units::literals::operator""_Pg (   
    unsigned long long x )
```

Definition at line 66 of file literals.h.

5.4.1.953 operator""_PGy() [1/2]

```
quantity< absorbed_dose , long double> boost::units::literals::operator""_PGy (   
    long double x )
```

Definition at line 91 of file literals.h.

5.4.1.954 operator""_PGy() [2/2]

```
quantity< absorbed_dose , unsigned long long> boost::units::literals::operator""_PGy (   
    unsigned long long x )
```

Definition at line 91 of file literals.h.

5.4.1.955 operator""_pGy() [1/2]

```
quantity< absorbed_dose , long double> boost::units::literals::operator""_pGy (   
    long double x )
```

Definition at line 91 of file literals.h.

5.4.1.956 operator""_pGy() [2/2]

```
quantity< absorbed_dose , unsigned long long> boost::units::literals::operator""_pGy (   
    unsigned long long x )
```

Definition at line 91 of file literals.h.

5.4.1.957 operator""_PH() [1/2]

```
quantity< inductance , unsigned long long> boost::units::literals::operator"" _PH (
    unsigned long long x )
```

Definition at line 86 of file literals.h.

5.4.1.958 operator""_PH() [2/2]

```
quantity< inductance , long double> boost::units::literals::operator"" _PH (
    long double x )
```

Definition at line 86 of file literals.h.

5.4.1.959 operator""_pH() [1/2]

```
quantity< inductance , long double> boost::units::literals::operator"" _pH (
    long double x )
```

Definition at line 86 of file literals.h.

5.4.1.960 operator""_pH() [2/2]

```
quantity< inductance , unsigned long long> boost::units::literals::operator"" _pH (
    unsigned long long x )
```

Definition at line 86 of file literals.h.

5.4.1.961 operator""_Ph() [1/2]

```
quantity< time , long double> boost::units::literals::operator"" _Ph (
    long double x )
```

Definition at line 95 of file literals.h.

5.4.1.962 operator""_Ph() [2/2]

```
quantity< time , unsigned long long> boost::units::literals::operator"" _Ph (
    unsigned long long x )
```

Definition at line 95 of file literals.h.

5.4.1.963 operator""_ph() [1/2]

```
quantity< time , long double> boost::units::literals::operator"" _ph (
    long double x )
```

Definition at line 95 of file literals.h.

5.4.1.964 operator""_ph() [2/2]

```
quantity< time , unsigned long long> boost::units::literals::operator""_ph (  
    unsigned long long x )
```

Definition at line 95 of file literals.h.

5.4.1.965 operator""_pHz() [1/2]

```
quantity< frequency , unsigned long long> boost::units::literals::operator""_pHz (   
    unsigned long long x )
```

Definition at line 72 of file literals.h.

5.4.1.966 operator""_PHz() [1/2]

```
quantity< frequency , unsigned long long> boost::units::literals::operator""_PHz (   
    unsigned long long x )
```

Definition at line 72 of file literals.h.

5.4.1.967 operator""_PHz() [2/2]

```
quantity< frequency , long double> boost::units::literals::operator""_PHz (   
    long double x )
```

Definition at line 72 of file literals.h.

5.4.1.968 operator""_pHz() [2/2]

```
quantity< frequency , long double> boost::units::literals::operator""_pHz (   
    long double x )
```

Definition at line 72 of file literals.h.

5.4.1.969 operator""_pJ() [1/2]

```
quantity< energy , unsigned long long> boost::units::literals::operator""_pJ (   
    unsigned long long x )
```

Definition at line 77 of file literals.h.

5.4.1.970 operator""_PJ() [1/2]

```
quantity< energy , unsigned long long> boost::units::literals::operator""_PJ (   
    unsigned long long x )
```

Definition at line 77 of file literals.h.

5.4.1.971 operator""_pj() [2/2]

```
quantity< energy , long double> boost::units::literals::operator"" _pj ( long double x )
```

Definition at line 77 of file literals.h.

5.4.1.972 operator""_PJ() [2/2]

```
quantity< energy , long double> boost::units::literals::operator"" _PJ ( long double x )
```

Definition at line 77 of file literals.h.

5.4.1.973 operator""_PK() [1/2]

```
quantity< temperature , long double> boost::units::literals::operator"" _PK ( long double x )
```

Definition at line 69 of file literals.h.

5.4.1.974 operator""_PK() [2/2]

```
quantity< temperature , unsigned long long> boost::units::literals::operator"" _PK ( unsigned long long x )
```

Definition at line 69 of file literals.h.

5.4.1.975 operator""_pK() [1/2]

```
quantity< temperature , long double> boost::units::literals::operator"" _pK ( long double x )
```

Definition at line 69 of file literals.h.

5.4.1.976 operator""_pK() [2/2]

```
quantity< temperature , unsigned long long> boost::units::literals::operator"" _pK ( unsigned long long x )
```

Definition at line 69 of file literals.h.

5.4.1.977 operator""_Pkat() [1/2]

```
quantity< catalytic_activity , long double> boost::units::literals::operator"" _Pkat ( long double x )
```

Definition at line 93 of file literals.h.

5.4.1.978 operator""_Pkat() [2/2]

```
quantity< catalytic_activity , unsigned long long> boost::units::literals::operator""_Pkat (  
    unsigned long long x )
```

Definition at line 93 of file literals.h.

5.4.1.979 operator""_pkat() [1/2]

```
quantity< catalytic_activity , long double> boost::units::literals::operator""_pkat (   
    long double x )
```

Definition at line 93 of file literals.h.

5.4.1.980 operator""_pkat() [2/2]

```
quantity< catalytic_activity , unsigned long long> boost::units::literals::operator""_pkat (   
    unsigned long long x )
```

Definition at line 93 of file literals.h.

5.4.1.981 operator""_Pl() [1/2]

```
quantity< volume , long double> boost::units::literals::operator""_Pl (   
    long double x )
```

Definition at line 98 of file literals.h.

5.4.1.982 operator""_Pl() [2/2]

```
quantity< volume , unsigned long long> boost::units::literals::operator""_Pl (   
    unsigned long long x )
```

Definition at line 98 of file literals.h.

5.4.1.983 operator""_pl() [1/2]

```
quantity< volume , long double> boost::units::literals::operator""_pl (   
    long double x )
```

Definition at line 98 of file literals.h.

5.4.1.984 operator""_pl() [2/2]

```
quantity< volume , unsigned long long> boost::units::literals::operator""_pl (   
    unsigned long long x )
```

Definition at line 98 of file literals.h.

5.4.1.985 operator""_PL() [1/2]

```
quantity< volume , unsigned long long> boost::units::literals::operator"" _PL (  
    unsigned long long x )
```

Definition at line 99 of file literals.h.

5.4.1.986 operator""_PL() [2/2]

```
quantity< volume , long double> boost::units::literals::operator"" _PL (  
    long double x )
```

Definition at line 99 of file literals.h.

5.4.1.987 operator""_pL() [1/2]

```
quantity< volume , unsigned long long> boost::units::literals::operator"" _pL (   
    unsigned long long x )
```

Definition at line 99 of file literals.h.

5.4.1.988 operator""_pL() [2/2]

```
quantity< volume , long double> boost::units::literals::operator"" _pL (   
    long double x )
```

Definition at line 99 of file literals.h.

5.4.1.989 operator""_plm() [1/2]

```
quantity< luminous_flux , long double> boost::units::literals::operator"" _plm (   
    long double x )
```

Definition at line 88 of file literals.h.

5.4.1.990 operator""_Plm() [1/2]

```
quantity< luminous_flux , long double> boost::units::literals::operator"" _Plm (   
    long double x )
```

Definition at line 88 of file literals.h.

5.4.1.991 operator""_Plm() [2/2]

```
quantity< luminous_flux , unsigned long long> boost::units::literals::operator"" _Plm (   
    unsigned long long x )
```

Definition at line 88 of file literals.h.

5.4.1.992 operator""_plm() [2/2]

```
quantity< luminous_flux , unsigned long long> boost::units::literals::operator""_plm (  
    unsigned long long x )
```

Definition at line 88 of file literals.h.

5.4.1.993 operator""_plx() [1/2]

```
quantity< illuminance , long double> boost::units::literals::operator""_plx (   
    long double x )
```

Definition at line 89 of file literals.h.

5.4.1.994 operator""_Plx() [1/2]

```
quantity< illuminance , long double> boost::units::literals::operator""_Plx (   
    long double x )
```

Definition at line 89 of file literals.h.

5.4.1.995 operator""_Plx() [2/2]

```
quantity< illuminance , unsigned long long> boost::units::literals::operator""_Plx (   
    unsigned long long x )
```

Definition at line 89 of file literals.h.

5.4.1.996 operator""_plx() [2/2]

```
quantity< illuminance , unsigned long long> boost::units::literals::operator""_plx (   
    unsigned long long x )
```

Definition at line 89 of file literals.h.

5.4.1.997 operator""_pm() [1/2]

```
quantity< length , long double> boost::units::literals::operator""_pm (   
    long double x )
```

Definition at line 65 of file literals.h.

5.4.1.998 operator""_Pm() [1/2]

```
quantity< length , long double> boost::units::literals::operator""_Pm (   
    long double x )
```

Definition at line 65 of file literals.h.

5.4.1.999 operator""_pm() [2/2]

```
quantity< length , unsigned long long> boost::units::literals::operator"" _pm (
    unsigned long long x )
```

Definition at line 65 of file literals.h.

5.4.1.1000 operator""_Pm() [2/2]

```
quantity< length , unsigned long long> boost::units::literals::operator"" _Pm (
    unsigned long long x )
```

Definition at line 65 of file literals.h.

5.4.1.1001 operator""_Pmin() [1/2]

```
quantity< time , long double> boost::units::literals::operator"" _Pmin (
    long double x )
```

Definition at line 94 of file literals.h.

5.4.1.1002 operator""_pmin() [1/2]

```
quantity< time , long double> boost::units::literals::operator"" _pmin (
    long double x )
```

Definition at line 94 of file literals.h.

5.4.1.1003 operator""_pmin() [2/2]

```
quantity< time , unsigned long long> boost::units::literals::operator"" _pmin (
    unsigned long long x )
```

Definition at line 94 of file literals.h.

5.4.1.1004 operator""_Pmin() [2/2]

```
quantity< time , unsigned long long> boost::units::literals::operator"" _Pmin (
    unsigned long long x )
```

Definition at line 94 of file literals.h.

5.4.1.1005 operator""_Pmol() [1/2]

```
quantity< amount , unsigned long long> boost::units::literals::operator"" _Pmol (
    unsigned long long x )
```

Definition at line 70 of file literals.h.

5.4.1.1006 operator""_Pmol() [2/2]

```
quantity< amount , long double> boost::units::literals::operator""_Pmol (  
    long double x )
```

Definition at line 70 of file literals.h.

5.4.1.1007 operator""_pmol() [1/2]

```
quantity< amount , unsigned long long> boost::units::literals::operator""_pmol (   
    unsigned long long x )
```

Definition at line 70 of file literals.h.

5.4.1.1008 operator""_pmol() [2/2]

```
quantity< amount , long double> boost::units::literals::operator""_pmol (   
    long double x )
```

Definition at line 70 of file literals.h.

5.4.1.1009 operator""_pN() [1/2]

```
quantity< force , unsigned long long> boost::units::literals::operator""_pN (   
    unsigned long long x )
```

Definition at line 75 of file literals.h.

5.4.1.1010 operator""_pN() [2/2]

```
quantity< force , long double> boost::units::literals::operator""_pN (   
    long double x )
```

Definition at line 75 of file literals.h.

5.4.1.1011 operator""_PN() [1/2]

```
quantity< force , long double> boost::units::literals::operator""_PN (   
    long double x )
```

Definition at line 75 of file literals.h.

5.4.1.1012 operator""_PN() [2/2]

```
quantity< force , unsigned long long> boost::units::literals::operator""_PN (   
    unsigned long long x )
```

Definition at line 75 of file literals.h.

5.4.1.1013 operator""_pohm() [1/2]

```
quantity< resistance , long double> boost::units::literals::operator"" _pohm (
    long double x )
```

Definition at line 82 of file literals.h.

5.4.1.1014 operator""_pohm() [2/2]

```
quantity< resistance , unsigned long long> boost::units::literals::operator"" _pohm (
    unsigned long long x )
```

Definition at line 82 of file literals.h.

5.4.1.1015 operator""_Pohm() [1/2]

```
quantity< resistance , long double> boost::units::literals::operator"" _Pohm (
    long double x )
```

Definition at line 82 of file literals.h.

5.4.1.1016 operator""_Pohm() [2/2]

```
quantity< resistance , unsigned long long> boost::units::literals::operator"" _Pohm (
    unsigned long long x )
```

Definition at line 82 of file literals.h.

5.4.1.1017 operator""_PPa() [1/2]

```
quantity< pressure , long double> boost::units::literals::operator"" _PPa (
    long double x )
```

Definition at line 76 of file literals.h.

5.4.1.1018 operator""_PPa() [1/2]

```
quantity< pressure , long double> boost::units::literals::operator"" _PPa (
    long double x )
```

Definition at line 76 of file literals.h.

5.4.1.1019 operator""_PPa() [2/2]

```
quantity< pressure , unsigned long long> boost::units::literals::operator"" _PPa (
    unsigned long long x )
```

Definition at line 76 of file literals.h.

5.4.1.1020 operator""_pPa() [2/2]

```
quantity< pressure , unsigned long long> boost::units::operator"" _pPa (  
    unsigned long long x )
```

Definition at line 76 of file literals.h.

5.4.1.1021 operator""_prad() [1/2]

```
quantity< plane_angle , unsigned long long> boost::units::operator"" _prad (   
    unsigned long long x )
```

Definition at line 73 of file literals.h.

5.4.1.1022 operator""_Prad() [1/2]

```
quantity< plane_angle , unsigned long long> boost::units::operator"" _Prad (   
    unsigned long long x )
```

Definition at line 73 of file literals.h.

5.4.1.1023 operator""_Prad() [2/2]

```
quantity< plane_angle , long double> boost::units::operator"" _Prad (   
    long double x )
```

Definition at line 73 of file literals.h.

5.4.1.1024 operator""_prad() [2/2]

```
quantity< plane_angle , long double> boost::units::operator"" _prad (   
    long double x )
```

Definition at line 73 of file literals.h.

5.4.1.1025 operator""_Ps() [1/2]

```
quantity< time , unsigned long long> boost::units::operator"" _Ps (   
    unsigned long long x )
```

Definition at line 67 of file literals.h.

5.4.1.1026 operator""_Ps() [2/2]

```
quantity< time , long double> boost::units::operator"" _Ps (   
    long double x )
```

Definition at line 67 of file literals.h.

5.4.1.1027 operator""_ps() [1/2]

```
quantity< time , unsigned long long> boost::units::literals::operator""_ps (  
    unsigned long long x )
```

Definition at line 67 of file literals.h.

5.4.1.1028 operator""_ps() [2/2]

```
quantity< time , long double> boost::units::literals::operator""_ps (   
    long double x )
```

Definition at line 67 of file literals.h.

5.4.1.1029 operator""_PS() [1/2]

```
quantity< conductance , long double> boost::units::literals::operator""_PS (   
    long double x )
```

Definition at line 83 of file literals.h.

5.4.1.1030 operator""_PS() [2/2]

```
quantity< conductance , unsigned long long> boost::units::literals::operator""_PS (   
    unsigned long long x )
```

Definition at line 83 of file literals.h.

5.4.1.1031 operator""_pS() [1/2]

```
quantity< conductance , long double> boost::units::literals::operator""_pS (   
    long double x )
```

Definition at line 83 of file literals.h.

5.4.1.1032 operator""_pS() [2/2]

```
quantity< conductance , unsigned long long> boost::units::literals::operator""_pS (   
    unsigned long long x )
```

Definition at line 83 of file literals.h.

5.4.1.1033 operator""_psr() [1/2]

```
quantity< solid_angle , unsigned long long> boost::units::literals::operator""_psr (   
    unsigned long long x )
```

Definition at line 74 of file literals.h.

5.4.1.1034 operator""_Psr() [1/2]

```
quantity< solid_angle , long double> boost::units::literals::operator""_Psr ( long double x )
```

Definition at line 74 of file literals.h.

5.4.1.1035 operator""_Psr() [2/2]

```
quantity< solid_angle , unsigned long long> boost::units::literals::operator""_Psr ( unsigned long long x )
```

Definition at line 74 of file literals.h.

5.4.1.1036 operator""_psr() [2/2]

```
quantity< solid_angle , long double> boost::units::literals::operator""_psr ( long double x )
```

Definition at line 74 of file literals.h.

5.4.1.1037 operator""_PSv() [1/2]

```
quantity< dose_equivalent , long double> boost::units::literals::operator""_PSv ( long double x )
```

Definition at line 92 of file literals.h.

5.4.1.1038 operator""_PSv() [2/2]

```
quantity< dose_equivalent , unsigned long long> boost::units::literals::operator""_PSv ( unsigned long long x )
```

Definition at line 92 of file literals.h.

5.4.1.1039 operator""_pSv() [1/2]

```
quantity< dose_equivalent , unsigned long long> boost::units::literals::operator""_pSv ( unsigned long long x )
```

Definition at line 92 of file literals.h.

5.4.1.1040 operator""_pSv() [2/2]

```
quantity< dose_equivalent , long double> boost::units::literals::operator""_pSv ( long double x )
```

Definition at line 92 of file literals.h.

5.4.1.1041 operator""_pT() [1/2]

```
quantity< magnetic_flux_density , long double> boost::units::literals::operator""_pT ( long double x )
```

Definition at line 85 of file literals.h.

5.4.1.1042 operator""_PT() [1/2]

```
quantity< magnetic_flux_density , long double> boost::units::literals::operator""_PT ( long double x )
```

Definition at line 85 of file literals.h.

5.4.1.1043 operator""_PT() [2/2]

```
quantity< magnetic_flux_density , unsigned long long> boost::units::literals::operator""_PT ( unsigned long long x )
```

Definition at line 85 of file literals.h.

5.4.1.1044 operator""_pT() [2/2]

```
quantity< magnetic_flux_density , unsigned long long> boost::units::literals::operator""_pT ( unsigned long long x )
```

Definition at line 85 of file literals.h.

5.4.1.1045 operator""_pt() [1/2]

```
quantity< mass , long double> boost::units::literals::operator""_pt ( long double x )
```

Definition at line 100 of file literals.h.

5.4.1.1046 operator""_Pt() [1/2]

```
quantity< mass , long double> boost::units::literals::operator""_Pt ( long double x )
```

Definition at line 100 of file literals.h.

5.4.1.1047 operator""_pt() [2/2]

```
quantity< mass , unsigned long long> boost::units::literals::operator""_pt ( unsigned long long x )
```

Definition at line 100 of file literals.h.

5.4.1.1048 operator""_Pt() [2/2]

```
quantity< mass , unsigned long long> boost::units::literals::operator""_Pt (  
    unsigned long long x )
```

Definition at line 100 of file literals.h.

5.4.1.1049 operator""_pV() [1/2]

```
quantity< electric_potential , long double> boost::units::literals::operator""_pV (   
    long double x )
```

Definition at line 80 of file literals.h.

5.4.1.1050 operator""_pV() [2/2]

```
quantity< electric_potential , unsigned long long> boost::units::literals::operator""_pV (   
    unsigned long long x )
```

Definition at line 80 of file literals.h.

5.4.1.1051 operator""_PV() [1/2]

```
quantity< electric_potential , unsigned long long> boost::units::literals::operator""_PV (   
    unsigned long long x )
```

Definition at line 80 of file literals.h.

5.4.1.1052 operator""_PV() [2/2]

```
quantity< electric_potential , long double> boost::units::literals::operator""_PV (   
    long double x )
```

Definition at line 80 of file literals.h.

5.4.1.1053 operator""_PW() [1/2]

```
quantity< power , unsigned long long> boost::units::literals::operator""_PW (   
    unsigned long long x )
```

Definition at line 78 of file literals.h.

5.4.1.1054 operator""_PW() [2/2]

```
quantity< power , long double> boost::units::literals::operator""_PW (   
    long double x )
```

Definition at line 78 of file literals.h.

5.4.1.1055 operator""_pW() [1/2]

```
quantity< power , long double> boost::units::literals::operator"" _pW (
    long double x )
```

Definition at line 78 of file literals.h.

5.4.1.1056 operator""_pW() [2/2]

```
quantity< power , unsigned long long> boost::units::literals::operator"" _pW (
    unsigned long long x )
```

Definition at line 78 of file literals.h.

5.4.1.1057 operator""_pWb() [1/2]

```
quantity< magnetic_flux , long double> boost::units::literals::operator"" _pWb (
    long double x )
```

Definition at line 84 of file literals.h.

5.4.1.1058 operator""_PWb() [1/2]

```
quantity< magnetic_flux , long double> boost::units::literals::operator"" _PWb (
    long double x )
```

Definition at line 84 of file literals.h.

5.4.1.1059 operator""_PWb() [2/2]

```
quantity< magnetic_flux , unsigned long long> boost::units::literals::operator"" _PWb (
    unsigned long long x )
```

Definition at line 84 of file literals.h.

5.4.1.1060 operator""_pWb() [2/2]

```
quantity< magnetic_flux , unsigned long long> boost::units::literals::operator"" _pWb (
    unsigned long long x )
```

Definition at line 84 of file literals.h.

5.4.1.1061 operator""_rad() [1/2]

```
quantity< plane_angle , unsigned long long> boost::units::literals::operator"" _rad (
    unsigned long long x )
```

Definition at line 73 of file literals.h.

5.4.1.1062 operator""_rad() [2/2]

```
quantity< plane_angle , long double> boost::units::literals::operator""_rad ( long double x )
```

Definition at line 73 of file literals.h.

5.4.1.1063 operator""_s() [1/2]

```
quantity< time , long double> boost::units::literals::operator""_s ( long double x )
```

Definition at line 67 of file literals.h.

5.4.1.1064 operator""_s() [2/2]

```
quantity< time , unsigned long long> boost::units::literals::operator""_s ( unsigned long long x )
```

Definition at line 67 of file literals.h.

5.4.1.1065 operator""_S() [1/2]

```
quantity< conductance , unsigned long long> boost::units::literals::operator""_S ( unsigned long long x )
```

Definition at line 83 of file literals.h.

5.4.1.1066 operator""_S() [2/2]

```
quantity< conductance , long double> boost::units::literals::operator""_S ( long double x )
```

Definition at line 83 of file literals.h.

5.4.1.1067 operator""_sr() [1/2]

```
quantity< solid_angle , long double> boost::units::literals::operator""_sr ( long double x )
```

Definition at line 74 of file literals.h.

5.4.1.1068 operator""_sr() [2/2]

```
quantity< solid_angle , unsigned long long> boost::units::literals::operator""_sr ( unsigned long long x )
```

Definition at line 74 of file literals.h.

5.4.1.1069 operator""_Sv() [1/2]

```
quantity< dose_equivalent , long double> boost::units::literals::operator""_Sv (
    long double x )
```

Definition at line 92 of file literals.h.

5.4.1.1070 operator""_Sv() [2/2]

```
quantity< dose_equivalent , unsigned long long> boost::units::literals::operator""_Sv (
    unsigned long long x )
```

Definition at line 92 of file literals.h.

5.4.1.1071 operator""_T() [1/2]

```
quantity< magnetic_flux_density , unsigned long long> boost::units::literals::operator""_T (
    unsigned long long x )
```

Definition at line 85 of file literals.h.

5.4.1.1072 operator""_T() [2/2]

```
quantity< magnetic_flux_density , long double> boost::units::literals::operator""_T (
    long double x )
```

Definition at line 85 of file literals.h.

5.4.1.1073 operator""_t() [1/2]

```
quantity< mass , long double> boost::units::literals::operator""_t (
    long double x )
```

Definition at line 100 of file literals.h.

5.4.1.1074 operator""_t() [2/2]

```
quantity< mass , unsigned long long> boost::units::literals::operator""_t (
    unsigned long long x )
```

Definition at line 100 of file literals.h.

5.4.1.1075 operator""_TA() [1/2]

```
quantity< current , long double> boost::units::literals::operator""_TA (
    long double x )
```

Definition at line 68 of file literals.h.

5.4.1.1076 operator""_TA() [2/2]

```
quantity< current , unsigned long long> boost::units::literals::operator""_TA (  
    unsigned long long x )
```

Definition at line 68 of file literals.h.

5.4.1.1077 operator""_TBq() [1/2]

```
quantity< activity , long double> boost::units::literals::operator""_TBq (   
    long double x )
```

Definition at line 90 of file literals.h.

5.4.1.1078 operator""_TBq() [2/2]

```
quantity< activity , unsigned long long> boost::units::literals::operator""_TBq (   
    unsigned long long x )
```

Definition at line 90 of file literals.h.

5.4.1.1079 operator""_TC() [1/2]

```
quantity< electric_charge , unsigned long long> boost::units::literals::operator""_TC (   
    unsigned long long x )
```

Definition at line 79 of file literals.h.

5.4.1.1080 operator""_TC() [2/2]

```
quantity< electric_charge , long double> boost::units::literals::operator""_TC (   
    long double x )
```

Definition at line 79 of file literals.h.

5.4.1.1081 operator""_Tcd() [1/2]

```
quantity< luminous_intensity , long double> boost::units::literals::operator""_Tcd (   
    long double x )
```

Definition at line 71 of file literals.h.

5.4.1.1082 operator""_Tcd() [2/2]

```
quantity< luminous_intensity , unsigned long long> boost::units::literals::operator""_Tcd (   
    unsigned long long x )
```

Definition at line 71 of file literals.h.

5.4.1.1083 operator""_Tday() [1/2]

```
quantity< time , unsigned long long> boost::units::literals::operator""_Tday (  
    unsigned long long x )
```

Definition at line 96 of file literals.h.

5.4.1.1084 operator""_Tday() [2/2]

```
quantity< time , long double> boost::units::literals::operator""_Tday (   
    long double x )
```

Definition at line 96 of file literals.h.

5.4.1.1085 operator""_Tdeg() [1/2]

```
quantity< plane_angle , long double> boost::units::literals::operator""_Tdeg (   
    long double x )
```

Definition at line 97 of file literals.h.

5.4.1.1086 operator""_Tdeg() [2/2]

```
quantity< plane_angle , unsigned long long> boost::units::literals::operator""_Tdeg (   
    unsigned long long x )
```

Definition at line 97 of file literals.h.

5.4.1.1087 operator""_TdegC() [1/2]

```
quantity< temperature , long double> boost::units::literals::operator""_TdegC (   
    long double x )
```

Definition at line 87 of file literals.h.

5.4.1.1088 operator""_TdegC() [2/2]

```
quantity< temperature , unsigned long long> boost::units::literals::operator""_TdegC (   
    unsigned long long x )
```

Definition at line 87 of file literals.h.

5.4.1.1089 operator""_TF() [1/2]

```
quantity< capacitance , unsigned long long> boost::units::literals::operator""_TF (   
    unsigned long long x )
```

Definition at line 81 of file literals.h.

5.4.1.1090 operator""_TF() [2/2]

```
quantity< capacitance , long double> boost::units::literals::operator""_TF ( long double x )
```

Definition at line 81 of file literals.h.

5.4.1.1091 operator""_Tg() [1/2]

```
quantity< mass , unsigned long long> boost::units::literals::operator""_Tg ( unsigned long long x )
```

Definition at line 66 of file literals.h.

5.4.1.1092 operator""_Tg() [2/2]

```
quantity< mass , long double> boost::units::literals::operator""_Tg ( long double x )
```

Definition at line 66 of file literals.h.

5.4.1.1093 operator""_TGy() [1/2]

```
quantity< absorbed_dose , long double> boost::units::literals::operator""_TGy ( long double x )
```

Definition at line 91 of file literals.h.

5.4.1.1094 operator""_TGy() [2/2]

```
quantity< absorbed_dose , unsigned long long> boost::units::literals::operator""_TGy ( unsigned long long x )
```

Definition at line 91 of file literals.h.

5.4.1.1095 operator""_TH() [1/2]

```
quantity< inductance , long double> boost::units::literals::operator""_TH ( long double x )
```

Definition at line 86 of file literals.h.

5.4.1.1096 operator""_TH() [2/2]

```
quantity< inductance , unsigned long long> boost::units::literals::operator""_TH ( unsigned long long x )
```

Definition at line 86 of file literals.h.

5.4.1.1097 operator""_Th() [1/2]

```
quantity< time , long double> boost::units::literals::operator""_Th (
    long double x )
```

Definition at line 95 of file literals.h.

5.4.1.1098 operator""_Th() [2/2]

```
quantity< time , unsigned long long> boost::units::literals::operator""_Th (
    unsigned long long x )
```

Definition at line 95 of file literals.h.

5.4.1.1099 operator""_THz() [1/2]

```
quantity< frequency , unsigned long long> boost::units::literals::operator""_THz (
    unsigned long long x )
```

Definition at line 72 of file literals.h.

5.4.1.1100 operator""_THz() [2/2]

```
quantity< frequency , long double> boost::units::literals::operator""_THz (
    long double x )
```

Definition at line 72 of file literals.h.

5.4.1.1101 operator""_TJ() [1/2]

```
quantity< energy , unsigned long long> boost::units::literals::operator""_TJ (
    unsigned long long x )
```

Definition at line 77 of file literals.h.

5.4.1.1102 operator""_TJ() [2/2]

```
quantity< energy , long double> boost::units::literals::operator""_TJ (
    long double x )
```

Definition at line 77 of file literals.h.

5.4.1.1103 operator""_TK() [1/2]

```
quantity< temperature , long double> boost::units::literals::operator""_TK (
    long double x )
```

Definition at line 69 of file literals.h.

5.4.1.1104 operator""_TK() [2/2]

```
quantity< temperature , unsigned long long> boost::units::literals::operator""_TK( unsigned long long x )
```

Definition at line 69 of file literals.h.

5.4.1.1105 operator""_Tkat() [1/2]

```
quantity< catalytic_activity , unsigned long long> boost::units::literals::operator""_Tkat( unsigned long long x )
```

Definition at line 93 of file literals.h.

5.4.1.1106 operator""_Tkat() [2/2]

```
quantity< catalytic_activity , long double> boost::units::literals::operator""_Tkat( long double x )
```

Definition at line 93 of file literals.h.

5.4.1.1107 operator""_Tl() [1/2]

```
quantity< volume , long double> boost::units::literals::operator""_Tl( long double x )
```

Definition at line 98 of file literals.h.

5.4.1.1108 operator""_Tl() [2/2]

```
quantity< volume , unsigned long long> boost::units::literals::operator""_Tl( unsigned long long x )
```

Definition at line 98 of file literals.h.

5.4.1.1109 operator""_TL() [1/2]

```
quantity< volume , long double> boost::units::literals::operator""_TL( long double x )
```

Definition at line 99 of file literals.h.

5.4.1.1110 operator""_TL() [2/2]

```
quantity< volume , unsigned long long> boost::units::literals::operator""_TL( unsigned long long x )
```

Definition at line 99 of file literals.h.

5.4.1.1111 operator""_Tlm() [1/2]

```
quantity< luminous_flux , long double> boost::units::literals::operator""_Tlm (
    long double x )
```

Definition at line 88 of file literals.h.

5.4.1.1112 operator""_Tlm() [2/2]

```
quantity< luminous_flux , unsigned long long> boost::units::literals::operator""_Tlm (
    unsigned long long x )
```

Definition at line 88 of file literals.h.

5.4.1.1113 operator""_Tlx() [1/2]

```
quantity< illuminance , long double> boost::units::literals::operator""_Tlx (
    long double x )
```

Definition at line 89 of file literals.h.

5.4.1.1114 operator""_Tlx() [2/2]

```
quantity< illuminance , unsigned long long> boost::units::literals::operator""_Tlx (
    unsigned long long x )
```

Definition at line 89 of file literals.h.

5.4.1.1115 operator""_Tm() [1/2]

```
quantity< length , long double> boost::units::literals::operator""_Tm (
    long double x )
```

Definition at line 65 of file literals.h.

5.4.1.1116 operator""_Tm() [2/2]

```
quantity< length , unsigned long long> boost::units::literals::operator""_Tm (
    unsigned long long x )
```

Definition at line 65 of file literals.h.

5.4.1.1117 operator""_Tmin() [1/2]

```
quantity< time , unsigned long long> boost::units::literals::operator""_Tmin (
    unsigned long long x )
```

Definition at line 94 of file literals.h.

5.4.1.1118 operator""_Tmin() [2/2]

```
quantity< time , long double> boost::units::literals::operator""_Tmin ( long double x )
```

Definition at line 94 of file literals.h.

5.4.1.1119 operator""_Tmol() [1/2]

```
quantity< amount , long double> boost::units::literals::operator""_Tmol ( long double x )
```

Definition at line 70 of file literals.h.

5.4.1.1120 operator""_Tmol() [2/2]

```
quantity< amount , unsigned long long> boost::units::literals::operator""_Tmol ( unsigned long long x )
```

Definition at line 70 of file literals.h.

5.4.1.1121 operator""_TN() [1/2]

```
quantity< force , long double> boost::units::literals::operator""_TN ( long double x )
```

Definition at line 75 of file literals.h.

5.4.1.1122 operator""_TN() [2/2]

```
quantity< force , unsigned long long> boost::units::literals::operator""_TN ( unsigned long long x )
```

Definition at line 75 of file literals.h.

5.4.1.1123 operator""_Tohm() [1/2]

```
quantity< resistance , long double> boost::units::literals::operator""_Tohm ( long double x )
```

Definition at line 82 of file literals.h.

5.4.1.1124 operator""_Tohm() [2/2]

```
quantity< resistance , unsigned long long> boost::units::literals::operator""_Tohm ( unsigned long long x )
```

Definition at line 82 of file literals.h.

5.4.1.1125 operator""_TPa() [1/2]

```
quantity< pressure , long double> boost::units::literals::operator""_TPa (
    long double x )
```

Definition at line 76 of file literals.h.

5.4.1.1126 operator""_TPa() [2/2]

```
quantity< pressure , unsigned long long> boost::units::literals::operator""_TPa (
    unsigned long long x )
```

Definition at line 76 of file literals.h.

5.4.1.1127 operator""_Trad() [1/2]

```
quantity< plane_angle , unsigned long long> boost::units::literals::operator""_Trad (
    unsigned long long x )
```

Definition at line 73 of file literals.h.

5.4.1.1128 operator""_Trad() [2/2]

```
quantity< plane_angle , long double> boost::units::literals::operator""_Trad (
    long double x )
```

Definition at line 73 of file literals.h.

5.4.1.1129 operator""_Ts() [1/2]

```
quantity< time , unsigned long long> boost::units::literals::operator""_Ts (
    unsigned long long x )
```

Definition at line 67 of file literals.h.

5.4.1.1130 operator""_Ts() [2/2]

```
quantity< time , long double> boost::units::literals::operator""_Ts (
    long double x )
```

Definition at line 67 of file literals.h.

5.4.1.1131 operator""_TS() [1/2]

```
quantity< conductance , unsigned long long> boost::units::literals::operator""_TS (
    unsigned long long x )
```

Definition at line 83 of file literals.h.

5.4.1.1132 operator""_TS() [2/2]

```
quantity< conductance , long double> boost::units::literals::operator""_TS ( long double x )
```

Definition at line 83 of file literals.h.

5.4.1.1133 operator""_Tsr() [1/2]

```
quantity< solid_angle , long double> boost::units::literals::operator""_Tsr ( long double x )
```

Definition at line 74 of file literals.h.

5.4.1.1134 operator""_Tsr() [2/2]

```
quantity< solid_angle , unsigned long long> boost::units::literals::operator""_Tsr ( unsigned long long x )
```

Definition at line 74 of file literals.h.

5.4.1.1135 operator""_TSv() [1/2]

```
quantity< dose_equivalent , long double> boost::units::literals::operator""_TSv ( long double x )
```

Definition at line 92 of file literals.h.

5.4.1.1136 operator""_TSv() [2/2]

```
quantity< dose_equivalent , unsigned long long> boost::units::literals::operator""_TSv ( unsigned long long x )
```

Definition at line 92 of file literals.h.

5.4.1.1137 operator""_TT() [1/2]

```
quantity< magnetic_flux_density , long double> boost::units::literals::operator""_TT ( long double x )
```

Definition at line 85 of file literals.h.

5.4.1.1138 operator""_TT() [2/2]

```
quantity< magnetic_flux_density , unsigned long long> boost::units::literals::operator""_TT ( unsigned long long x )
```

Definition at line 85 of file literals.h.

5.4.1.1139 operator""_Tt() [1/2]

```
quantity< mass , long double> boost::units::literals::operator""_Tt (
    long double x )
```

Definition at line 100 of file literals.h.

5.4.1.1140 operator""_Tt() [2/2]

```
quantity< mass , unsigned long long> boost::units::literals::operator""_Tt (
    unsigned long long x )
```

Definition at line 100 of file literals.h.

5.4.1.1141 operator""_TV() [1/2]

```
quantity< electric_potential , unsigned long long> boost::units::literals::operator""_TV (
    unsigned long long x )
```

Definition at line 80 of file literals.h.

5.4.1.1142 operator""_TV() [2/2]

```
quantity< electric_potential , long double> boost::units::literals::operator""_TV (
    long double x )
```

Definition at line 80 of file literals.h.

5.4.1.1143 operator""_TW() [1/2]

```
quantity< power , unsigned long long> boost::units::literals::operator""_TW (
    unsigned long long x )
```

Definition at line 78 of file literals.h.

5.4.1.1144 operator""_TW() [2/2]

```
quantity< power , long double> boost::units::literals::operator""_TW (
    long double x )
```

Definition at line 78 of file literals.h.

5.4.1.1145 operator""_TWb() [1/2]

```
quantity< magnetic_flux , long double> boost::units::literals::operator""_TWb (
    long double x )
```

Definition at line 84 of file literals.h.

5.4.1.1146 operator""_TWb() [2/2]

```
quantity< magnetic_flux , unsigned long long> boost::units::literals::operator""_TWb (  
    unsigned long long x )
```

Definition at line 84 of file literals.h.

5.4.1.1147 operator""_uA() [1/2]

```
quantity< current , unsigned long long> boost::units::literals::operator""_uA (   
    unsigned long long x )
```

Definition at line 68 of file literals.h.

5.4.1.1148 operator""_uA() [2/2]

```
quantity< current , long double> boost::units::literals::operator""_uA (   
    long double x )
```

Definition at line 68 of file literals.h.

5.4.1.1149 operator""_uBq() [1/2]

```
quantity< activity , unsigned long long> boost::units::literals::operator""_uBq (   
    unsigned long long x )
```

Definition at line 90 of file literals.h.

5.4.1.1150 operator""_uBq() [2/2]

```
quantity< activity , long double> boost::units::literals::operator""_uBq (   
    long double x )
```

Definition at line 90 of file literals.h.

5.4.1.1151 operator""_uC() [1/2]

```
quantity< electric_charge , long double> boost::units::literals::operator""_uC (   
    long double x )
```

Definition at line 79 of file literals.h.

5.4.1.1152 operator""_uC() [2/2]

```
quantity< electric_charge , unsigned long long> boost::units::literals::operator""_uC (   
    unsigned long long x )
```

Definition at line 79 of file literals.h.

5.4.1.1153 operator""_ucd() [1/2]

```
quantity< luminous_intensity , unsigned long long> boost::units::literals::operator"" _ucd (
    unsigned long long x )
```

Definition at line 71 of file literals.h.

5.4.1.1154 operator""_ucd() [2/2]

```
quantity< luminous_intensity , long double> boost::units::literals::operator"" _ucd (
    long double x )
```

Definition at line 71 of file literals.h.

5.4.1.1155 operator""_uday() [1/2]

```
quantity< time , long double> boost::units::literals::operator"" _uday (
    long double x )
```

Definition at line 96 of file literals.h.

5.4.1.1156 operator""_uday() [2/2]

```
quantity< time , unsigned long long> boost::units::literals::operator"" _uday (
    unsigned long long x )
```

Definition at line 96 of file literals.h.

5.4.1.1157 operator""_udeg() [1/2]

```
quantity< plane_angle , unsigned long long> boost::units::literals::operator"" _udeg (
    unsigned long long x )
```

Definition at line 97 of file literals.h.

5.4.1.1158 operator""_udeg() [2/2]

```
quantity< plane_angle , long double> boost::units::literals::operator"" _udeg (
    long double x )
```

Definition at line 97 of file literals.h.

5.4.1.1159 operator""_udegC() [1/2]

```
quantity< temperature , unsigned long long> boost::units::literals::operator"" _udegC (
    unsigned long long x )
```

Definition at line 87 of file literals.h.

5.4.1.1160 operator""_udegC() [2/2]

```
quantity< temperature , long double> boost::units::literals::operator""_udegC ( long double x )
```

Definition at line 87 of file literals.h.

5.4.1.1161 operator""_uF() [1/2]

```
quantity< capacitance , long double> boost::units::literals::operator""_uF ( long double x )
```

Definition at line 81 of file literals.h.

5.4.1.1162 operator""_uF() [2/2]

```
quantity< capacitance , unsigned long long> boost::units::literals::operator""_uF ( unsigned long long x )
```

Definition at line 81 of file literals.h.

5.4.1.1163 operator""_ug() [1/2]

```
quantity< mass , unsigned long long> boost::units::literals::operator""_ug ( unsigned long long x )
```

Definition at line 66 of file literals.h.

5.4.1.1164 operator""_ug() [2/2]

```
quantity< mass , long double> boost::units::literals::operator""_ug ( long double x )
```

Definition at line 66 of file literals.h.

5.4.1.1165 operator""_uGy() [1/2]

```
quantity< absorbed_dose , long double> boost::units::literals::operator""_uGy ( long double x )
```

Definition at line 91 of file literals.h.

5.4.1.1166 operator""_uGy() [2/2]

```
quantity< absorbed_dose , unsigned long long> boost::units::literals::operator""_uGy ( unsigned long long x )
```

Definition at line 91 of file literals.h.

5.4.1.1167 operator""_uH() [1/2]

```
quantity< inductance , long double> boost::units::literals::operator""_uH ( long double x )
```

Definition at line 86 of file literals.h.

5.4.1.1168 operator""_uH() [2/2]

```
quantity< inductance , unsigned long long> boost::units::literals::operator""_uH ( unsigned long long x )
```

Definition at line 86 of file literals.h.

5.4.1.1169 operator""_uh() [1/2]

```
quantity< time , long double> boost::units::literals::operator""_uh ( long double x )
```

Definition at line 95 of file literals.h.

5.4.1.1170 operator""_uh() [2/2]

```
quantity< time , unsigned long long> boost::units::literals::operator""_uh ( unsigned long long x )
```

Definition at line 95 of file literals.h.

5.4.1.1171 operator""_uHz() [1/2]

```
quantity< frequency , long double> boost::units::literals::operator""_uHz ( long double x )
```

Definition at line 72 of file literals.h.

5.4.1.1172 operator""_uHz() [2/2]

```
quantity< frequency , unsigned long long> boost::units::literals::operator""_uHz ( unsigned long long x )
```

Definition at line 72 of file literals.h.

5.4.1.1173 operator""_uJ() [1/2]

```
quantity< energy , unsigned long long> boost::units::literals::operator""_uJ ( unsigned long long x )
```

Definition at line 77 of file literals.h.

5.4.1.1174 operator""_uJ() [2/2]

```
quantity< energy , long double> boost::units::literals::operator""_uJ ( long double x )
```

Definition at line 77 of file literals.h.

5.4.1.1175 operator""_uK() [1/2]

```
quantity< temperature , unsigned long long> boost::units::literals::operator""_uK ( unsigned long long x )
```

Definition at line 69 of file literals.h.

5.4.1.1176 operator""_uK() [2/2]

```
quantity< temperature , long double> boost::units::literals::operator""_uK ( long double x )
```

Definition at line 69 of file literals.h.

5.4.1.1177 operator""_ukat() [1/2]

```
quantity< catalytic_activity , unsigned long long> boost::units::literals::operator""_ukat ( unsigned long long x )
```

Definition at line 93 of file literals.h.

5.4.1.1178 operator""_ukat() [2/2]

```
quantity< catalytic_activity , long double> boost::units::literals::operator""_ukat ( long double x )
```

Definition at line 93 of file literals.h.

5.4.1.1179 operator""_ul() [1/2]

```
quantity< volume , unsigned long long> boost::units::literals::operator""_ul ( unsigned long long x )
```

Definition at line 98 of file literals.h.

5.4.1.1180 operator""_ul() [2/2]

```
quantity< volume , long double> boost::units::literals::operator""_ul ( long double x )
```

Definition at line 98 of file literals.h.

5.4.1.1181 operator""_uL() [1/2]

```
quantity< volume , unsigned long long> boost::units::literals::operator""_uL (  
    unsigned long long x )
```

Definition at line 99 of file literals.h.

5.4.1.1182 operator""_uL() [2/2]

```
quantity< volume , long double> boost::units::literals::operator""_uL (   
    long double x )
```

Definition at line 99 of file literals.h.

5.4.1.1183 operator""_ulm() [1/2]

```
quantity< luminous_flux , long double> boost::units::literals::operator""_ulm (   
    long double x )
```

Definition at line 88 of file literals.h.

5.4.1.1184 operator""_ulm() [2/2]

```
quantity< luminous_flux , unsigned long long> boost::units::literals::operator""_ulm (   
    unsigned long long x )
```

Definition at line 88 of file literals.h.

5.4.1.1185 operator""_ulx() [1/2]

```
quantity< illuminance , unsigned long long> boost::units::literals::operator""_ulx (   
    unsigned long long x )
```

Definition at line 89 of file literals.h.

5.4.1.1186 operator""_ulx() [2/2]

```
quantity< illuminance , long double> boost::units::literals::operator""_ulx (   
    long double x )
```

Definition at line 89 of file literals.h.

5.4.1.1187 operator""_um() [1/2]

```
quantity< length , long double> boost::units::literals::operator""_um (   
    long double x )
```

Definition at line 65 of file literals.h.

5.4.1.1188 operator""_um() [2/2]

```
quantity< length , unsigned long long> boost::units::literals::operator""_um (  
    unsigned long long x )
```

Definition at line 65 of file literals.h.

5.4.1.1189 operator""_umin() [1/2]

```
quantity< time , long double> boost::units::literals::operator""_umin (   
    long double x )
```

Definition at line 94 of file literals.h.

5.4.1.1190 operator""_umin() [2/2]

```
quantity< time , unsigned long long> boost::units::literals::operator""_umin (   
    unsigned long long x )
```

Definition at line 94 of file literals.h.

5.4.1.1191 operator""_umol() [1/2]

```
quantity< amount , unsigned long long> boost::units::literals::operator""_umol (   
    unsigned long long x )
```

Definition at line 70 of file literals.h.

5.4.1.1192 operator""_umol() [2/2]

```
quantity< amount , long double> boost::units::literals::operator""_umol (   
    long double x )
```

Definition at line 70 of file literals.h.

5.4.1.1193 operator""_uN() [1/2]

```
quantity< force , unsigned long long> boost::units::literals::operator""_uN (   
    unsigned long long x )
```

Definition at line 75 of file literals.h.

5.4.1.1194 operator""_uN() [2/2]

```
quantity< force , long double> boost::units::literals::operator""_uN (   
    long double x )
```

Definition at line 75 of file literals.h.

5.4.1.1195 operator""_uohm() [1/2]

```
quantity< resistance , long double> boost::units::literals::operator""_uohm ( long double x )
```

Definition at line 82 of file literals.h.

5.4.1.1196 operator""_uohm() [2/2]

```
quantity< resistance , unsigned long long> boost::units::literals::operator""_uohm ( unsigned long long x )
```

Definition at line 82 of file literals.h.

5.4.1.1197 operator""_uPa() [1/2]

```
quantity< pressure , long double> boost::units::literals::operator""_uPa ( long double x )
```

Definition at line 76 of file literals.h.

5.4.1.1198 operator""_uPa() [2/2]

```
quantity< pressure , unsigned long long> boost::units::literals::operator""_uPa ( unsigned long long x )
```

Definition at line 76 of file literals.h.

5.4.1.1199 operator""_urad() [1/2]

```
quantity< plane_angle , long double> boost::units::literals::operator""_urad ( long double x )
```

Definition at line 73 of file literals.h.

5.4.1.1200 operator""_urad() [2/2]

```
quantity< plane_angle , unsigned long long> boost::units::literals::operator""_urad ( unsigned long long x )
```

Definition at line 73 of file literals.h.

5.4.1.1201 operator""_us() [1/2]

```
quantity< time , unsigned long long> boost::units::literals::operator""_us ( unsigned long long x )
```

Definition at line 67 of file literals.h.

5.4.1.1202 operator""_us() [2/2]

```
quantity< time , long double> boost::units::literals::operator""_us (  
    long double x )
```

Definition at line 67 of file literals.h.

5.4.1.1203 operator""_uS() [1/2]

```
quantity< conductance , long double> boost::units::literals::operator""_uS (   
    long double x )
```

Definition at line 83 of file literals.h.

5.4.1.1204 operator""_uS() [2/2]

```
quantity< conductance , unsigned long long> boost::units::literals::operator""_uS (   
    unsigned long long x )
```

Definition at line 83 of file literals.h.

5.4.1.1205 operator""_usr() [1/2]

```
quantity< solid_angle , unsigned long long> boost::units::literals::operator""_usr (   
    unsigned long long x )
```

Definition at line 74 of file literals.h.

5.4.1.1206 operator""_usr() [2/2]

```
quantity< solid_angle , long double> boost::units::literals::operator""_usr (   
    long double x )
```

Definition at line 74 of file literals.h.

5.4.1.1207 operator""_uSv() [1/2]

```
quantity< dose_equivalent , unsigned long long> boost::units::literals::operator""_uSv (   
    unsigned long long x )
```

Definition at line 92 of file literals.h.

5.4.1.1208 operator""_uSv() [2/2]

```
quantity< dose_equivalent , long double> boost::units::literals::operator""_uSv (   
    long double x )
```

Definition at line 92 of file literals.h.

5.4.1.1209 operator""_uT() [1/2]

```
quantity< magnetic_flux_density , long double> boost::units::literals::operator"" _uT (
    long double x )
```

Definition at line 85 of file literals.h.

5.4.1.1210 operator""_uT() [2/2]

```
quantity< magnetic_flux_density , unsigned long long> boost::units::literals::operator"" _uT (
    unsigned long long x )
```

Definition at line 85 of file literals.h.

5.4.1.1211 operator""_ut() [1/2]

```
quantity< mass , long double> boost::units::literals::operator"" _ut (
    long double x )
```

Definition at line 100 of file literals.h.

5.4.1.1212 operator""_ut() [2/2]

```
quantity< mass , unsigned long long> boost::units::literals::operator"" _ut (
    unsigned long long x )
```

Definition at line 100 of file literals.h.

5.4.1.1213 operator""_uV() [1/2]

```
quantity< electric_potential , unsigned long long> boost::units::literals::operator"" _uV (
    unsigned long long x )
```

Definition at line 80 of file literals.h.

5.4.1.1214 operator""_uV() [2/2]

```
quantity< electric_potential , long double> boost::units::literals::operator"" _uV (
    long double x )
```

Definition at line 80 of file literals.h.

5.4.1.1215 operator""_uW() [1/2]

```
quantity< power , long double> boost::units::literals::operator"" _uW (
    long double x )
```

Definition at line 78 of file literals.h.

5.4.1.1216 operator""_uW() [2/2]

```
quantity< power , unsigned long long> boost::units::literals::operator""_uW( unsigned long long x )
```

Definition at line 78 of file literals.h.

5.4.1.1217 operator""_uWb() [1/2]

```
quantity< magnetic_flux , long double> boost::units::literals::operator""_uWb( long double x )
```

Definition at line 84 of file literals.h.

5.4.1.1218 operator""_uWb() [2/2]

```
quantity< magnetic_flux , unsigned long long> boost::units::literals::operator""_uWb( unsigned long long x )
```

Definition at line 84 of file literals.h.

5.4.1.1219 operator""_V() [1/2]

```
quantity< electric_potential , long double> boost::units::literals::operator""_V( long double x )
```

Definition at line 80 of file literals.h.

5.4.1.1220 operator""_V() [2/2]

```
quantity< electric_potential , unsigned long long> boost::units::literals::operator""_V( unsigned long long x )
```

Definition at line 80 of file literals.h.

5.4.1.1221 operator""_W() [1/2]

```
quantity< power , long double> boost::units::literals::operator""_W( long double x )
```

Definition at line 78 of file literals.h.

5.4.1.1222 operator""_W() [2/2]

```
quantity< power , unsigned long long> boost::units::literals::operator""_W( unsigned long long x )
```

Definition at line 78 of file literals.h.

5.4.1.1223 operator""_Wb() [1/2]

```
quantity< magnetic_flux , long double> boost::units::literals::operator"" _Wb (
    long double x )
```

Definition at line 84 of file literals.h.

5.4.1.1224 operator""_Wb() [2/2]

```
quantity< magnetic_flux , unsigned long long> boost::units::literals::operator"" _Wb (
    unsigned long long x )
```

Definition at line 84 of file literals.h.

5.4.1.1225 operator""_YA() [1/2]

```
quantity< current , unsigned long long> boost::units::literals::operator"" _YA (
    unsigned long long x )
```

Definition at line 68 of file literals.h.

5.4.1.1226 operator""_yA() [1/2]

```
quantity< current , long double> boost::units::literals::operator"" _yA (
    long double x )
```

Definition at line 68 of file literals.h.

5.4.1.1227 operator""_YA() [2/2]

```
quantity< current , long double> boost::units::literals::operator"" _YA (
    long double x )
```

Definition at line 68 of file literals.h.

5.4.1.1228 operator""_yA() [2/2]

```
quantity< current , unsigned long long> boost::units::literals::operator"" _yA (
    unsigned long long x )
```

Definition at line 68 of file literals.h.

5.4.1.1229 operator""_YBq() [1/2]

```
quantity< activity , long double> boost::units::literals::operator"" _YBq (
    long double x )
```

Definition at line 90 of file literals.h.

5.4.1.1230 operator""_yBq() [1/2]

```
quantity< activity , long double> boost::units::literals::operator""_yBq ( long double x )
```

Definition at line 90 of file literals.h.

5.4.1.1231 operator""_YBq() [2/2]

```
quantity< activity , unsigned long long> boost::units::literals::operator""_YBq ( unsigned long long x )
```

Definition at line 90 of file literals.h.

5.4.1.1232 operator""_yBq() [2/2]

```
quantity< activity , unsigned long long> boost::units::literals::operator""_yBq ( unsigned long long x )
```

Definition at line 90 of file literals.h.

5.4.1.1233 operator""_YC() [1/2]

```
quantity< electric_charge , long double> boost::units::literals::operator""_YC ( long double x )
```

Definition at line 79 of file literals.h.

5.4.1.1234 operator""_YC() [2/2]

```
quantity< electric_charge , unsigned long long> boost::units::literals::operator""_YC ( unsigned long long x )
```

Definition at line 79 of file literals.h.

5.4.1.1235 operator""_yC() [1/2]

```
quantity< electric_charge , unsigned long long> boost::units::literals::operator""_yC ( unsigned long long x )
```

Definition at line 79 of file literals.h.

5.4.1.1236 operator""_yC() [2/2]

```
quantity< electric_charge , long double> boost::units::literals::operator""_yC ( long double x )
```

Definition at line 79 of file literals.h.

5.4.1.1237 operator""_ycd() [1/2]

```
quantity< luminous_intensity , long double> boost::units::literals::operator"" _ycd ( long double x )
```

Definition at line 71 of file literals.h.

5.4.1.1238 operator""_ycd() [2/2]

```
quantity< luminous_intensity , unsigned long long> boost::units::literals::operator"" _ycd ( unsigned long long x )
```

Definition at line 71 of file literals.h.

5.4.1.1239 operator""_Ycd() [1/2]

```
quantity< luminous_intensity , long double> boost::units::literals::operator"" _Ycd ( long double x )
```

Definition at line 71 of file literals.h.

5.4.1.1240 operator""_Ycd() [2/2]

```
quantity< luminous_intensity , unsigned long long> boost::units::literals::operator"" _Ycd ( unsigned long long x )
```

Definition at line 71 of file literals.h.

5.4.1.1241 operator""_Yday() [1/2]

```
quantity< time , unsigned long long> boost::units::literals::operator"" _Yday ( unsigned long long x )
```

Definition at line 96 of file literals.h.

5.4.1.1242 operator""_Yday() [1/2]

```
quantity< time , long double> boost::units::literals::operator"" _Yday ( long double x )
```

Definition at line 96 of file literals.h.

5.4.1.1243 operator""_yday() [2/2]

```
quantity< time , unsigned long long> boost::units::literals::operator"" _yday ( unsigned long long x )
```

Definition at line 96 of file literals.h.

5.4.1.1244 operator""_Yday() [2/2]

```
quantity< time , long double> boost::units::literals::operator""_Yday ( long double x )
```

Definition at line 96 of file literals.h.

5.4.1.1245 operator""_Ydeg() [1/2]

```
quantity< plane_angle , unsigned long long> boost::units::literals::operator""_Ydeg ( unsigned long long x )
```

Definition at line 97 of file literals.h.

5.4.1.1246 operator""_ydeg() [1/2]

```
quantity< plane_angle , unsigned long long> boost::units::literals::operator""_ydeg ( unsigned long long x )
```

Definition at line 97 of file literals.h.

5.4.1.1247 operator""_Ydeg() [2/2]

```
quantity< plane_angle , long double> boost::units::literals::operator""_Ydeg ( long double x )
```

Definition at line 97 of file literals.h.

5.4.1.1248 operator""_ydeg() [2/2]

```
quantity< plane_angle , long double> boost::units::literals::operator""_ydeg ( long double x )
```

Definition at line 97 of file literals.h.

5.4.1.1249 operator""_ydegC() [1/2]

```
quantity< temperature , unsigned long long> boost::units::literals::operator""_ydegC ( unsigned long long x )
```

Definition at line 87 of file literals.h.

5.4.1.1250 operator""_YdegC() [1/2]

```
quantity< temperature , unsigned long long> boost::units::literals::operator""_YdegC ( unsigned long long x )
```

Definition at line 87 of file literals.h.

5.4.1.1251 operator""_YdegC() [2/2]

```
quantity< temperature , long double> boost::units::literals::operator""_YdegC ( long double x )
```

Definition at line 87 of file literals.h.

5.4.1.1252 operator""_ydegC() [2/2]

```
quantity< temperature , long double> boost::units::literals::operator""_ydegC ( long double x )
```

Definition at line 87 of file literals.h.

5.4.1.1253 operator""_YF() [1/2]

```
quantity< capacitance , unsigned long long> boost::units::literals::operator""_YF ( unsigned long long x )
```

Definition at line 81 of file literals.h.

5.4.1.1254 operator""_yF() [2/2]

```
quantity< capacitance , long double> boost::units::literals::operator""_yF ( long double x )
```

Definition at line 81 of file literals.h.

5.4.1.1255 operator""_yF() [1/2]

```
quantity< capacitance , unsigned long long> boost::units::literals::operator""_yF ( unsigned long long x )
```

Definition at line 81 of file literals.h.

5.4.1.1256 operator""_yF() [2/2]

```
quantity< capacitance , long double> boost::units::literals::operator""_yF ( long double x )
```

Definition at line 81 of file literals.h.

5.4.1.1257 operator""_yg() [1/2]

```
quantity< mass , long double> boost::units::literals::operator""_yg ( long double x )
```

Definition at line 66 of file literals.h.

5.4.1.1258 operator""_Yg() [1/2]

```
quantity< mass , unsigned long long> boost::units::literals::operator""_Yg (  
    unsigned long long x )
```

Definition at line 66 of file literals.h.

5.4.1.1259 operator""_Yg() [2/2]

```
quantity< mass , long double> boost::units::literals::operator""_Yg (   
    long double x )
```

Definition at line 66 of file literals.h.

5.4.1.1260 operator""_yg() [2/2]

```
quantity< mass , unsigned long long> boost::units::literals::operator""_yg (   
    unsigned long long x )
```

Definition at line 66 of file literals.h.

5.4.1.1261 operator""_yGy() [1/2]

```
quantity< absorbed_dose , long double> boost::units::literals::operator""_yGy (   
    long double x )
```

Definition at line 91 of file literals.h.

5.4.1.1262 operator""_yGy() [2/2]

```
quantity< absorbed_dose , unsigned long long> boost::units::literals::operator""_yGy (   
    unsigned long long x )
```

Definition at line 91 of file literals.h.

5.4.1.1263 operator""_YGy() [1/2]

```
quantity< absorbed_dose , unsigned long long> boost::units::literals::operator""_YGy (   
    unsigned long long x )
```

Definition at line 91 of file literals.h.

5.4.1.1264 operator""_YGy() [2/2]

```
quantity< absorbed_dose , long double> boost::units::literals::operator""_YGy (   
    long double x )
```

Definition at line 91 of file literals.h.

5.4.1.1265 operator""_yH() [1/2]

```
quantity< inductance , unsigned long long> boost::units::literals::operator"" _yH (  
    unsigned long long x )
```

Definition at line 86 of file literals.h.

5.4.1.1266 operator""_YH() [1/2]

```
quantity< inductance , long double> boost::units::literals::operator"" _YH (   
    long double x )
```

Definition at line 86 of file literals.h.

5.4.1.1267 operator""_YH() [2/2]

```
quantity< inductance , unsigned long long> boost::units::literals::operator"" _YH (   
    unsigned long long x )
```

Definition at line 86 of file literals.h.

5.4.1.1268 operator""_yH() [2/2]

```
quantity< inductance , long double> boost::units::literals::operator"" _yH (   
    long double x )
```

Definition at line 86 of file literals.h.

5.4.1.1269 operator""_Yh() [1/2]

```
quantity< time , unsigned long long> boost::units::literals::operator"" _Yh (   
    unsigned long long x )
```

Definition at line 95 of file literals.h.

5.4.1.1270 operator""_yh() [1/2]

```
quantity< time , unsigned long long> boost::units::literals::operator"" _yh (   
    unsigned long long x )
```

Definition at line 95 of file literals.h.

5.4.1.1271 operator""_yh() [2/2]

```
quantity< time , long double> boost::units::literals::operator"" _yh (   
    long double x )
```

Definition at line 95 of file literals.h.

5.4.1.1272 operator""_Yh() [2/2]

```
quantity< time , long double> boost::units::literals::operator""_Yh ( long double x )
```

Definition at line 95 of file literals.h.

5.4.1.1273 operator""_YHz() [1/2]

```
quantity< frequency , long double> boost::units::literals::operator""_YHz ( long double x )
```

Definition at line 72 of file literals.h.

5.4.1.1274 operator""_yHz() [1/2]

```
quantity< frequency , long double> boost::units::literals::operator""_yHz ( long double x )
```

Definition at line 72 of file literals.h.

5.4.1.1275 operator""_yHz() [2/2]

```
quantity< frequency , unsigned long long> boost::units::literals::operator""_yHz ( unsigned long long x )
```

Definition at line 72 of file literals.h.

5.4.1.1276 operator""_YHz() [2/2]

```
quantity< frequency , unsigned long long> boost::units::literals::operator""_YHz ( unsigned long long x )
```

Definition at line 72 of file literals.h.

5.4.1.1277 operator""_yJ() [1/2]

```
quantity< energy , unsigned long long> boost::units::literals::operator""_yJ ( unsigned long long x )
```

Definition at line 77 of file literals.h.

5.4.1.1278 operator""_yJ() [2/2]

```
quantity< energy , long double> boost::units::literals::operator""_yJ ( long double x )
```

Definition at line 77 of file literals.h.

5.4.1.1279 operator""_YJ() [1/2]

```
quantity< energy , unsigned long long> boost::units::literals::operator"" _YJ (
    unsigned long long x )
```

Definition at line 77 of file literals.h.

5.4.1.1280 operator""_YJ() [2/2]

```
quantity< energy , long double> boost::units::literals::operator"" _YJ (
    long double x )
```

Definition at line 77 of file literals.h.

5.4.1.1281 operator""_YK() [1/2]

```
quantity< temperature , unsigned long long> boost::units::literals::operator"" _YK (
    unsigned long long x )
```

Definition at line 69 of file literals.h.

5.4.1.1282 operator""_yK() [1/2]

```
quantity< temperature , long double> boost::units::literals::operator"" _yK (
    long double x )
```

Definition at line 69 of file literals.h.

5.4.1.1283 operator""_yK() [2/2]

```
quantity< temperature , unsigned long long> boost::units::literals::operator"" _yK (
    unsigned long long x )
```

Definition at line 69 of file literals.h.

5.4.1.1284 operator""_YK() [2/2]

```
quantity< temperature , long double> boost::units::literals::operator"" _YK (
    long double x )
```

Definition at line 69 of file literals.h.

5.4.1.1285 operator""_ykat() [1/2]

```
quantity< catalytic_activity , unsigned long long> boost::units::literals::operator"" _ykat (
    unsigned long long x )
```

Definition at line 93 of file literals.h.

5.4.1.1286 operator""_Ykat() [1/2]

```
quantity< catalytic_activity , unsigned long long> boost::units::literals::operator""_Ykat (  
    unsigned long long x )
```

Definition at line 93 of file literals.h.

5.4.1.1287 operator""_ykat() [2/2]

```
quantity< catalytic_activity , long double> boost::units::literals::operator""_ykat (   
    long double x )
```

Definition at line 93 of file literals.h.

5.4.1.1288 operator""_Ykat() [2/2]

```
quantity< catalytic_activity , long double> boost::units::literals::operator""_Ykat (   
    long double x )
```

Definition at line 93 of file literals.h.

5.4.1.1289 operator""_Yl() [1/2]

```
quantity< volume , unsigned long long> boost::units::literals::operator""_Yl (   
    unsigned long long x )
```

Definition at line 98 of file literals.h.

5.4.1.1290 operator""_yl() [1/2]

```
quantity< volume , long double> boost::units::literals::operator""_yl (   
    long double x )
```

Definition at line 98 of file literals.h.

5.4.1.1291 operator""_Yl() [2/2]

```
quantity< volume , unsigned long long> boost::units::literals::operator""_Yl (   
    unsigned long long x )
```

Definition at line 98 of file literals.h.

5.4.1.1292 operator""_Yl() [2/2]

```
quantity< volume , long double> boost::units::literals::operator""_Yl (   
    long double x )
```

Definition at line 98 of file literals.h.

5.4.1.1293 operator""_yL() [1/2]

```
quantity< volume , long double> boost::units::literals::operator"" _yL ( long double x )
```

Definition at line 99 of file literals.h.

5.4.1.1294 operator""_YL() [1/2]

```
quantity< volume , unsigned long long> boost::units::literals::operator"" _YL ( unsigned long long x )
```

Definition at line 99 of file literals.h.

5.4.1.1295 operator""_YL() [2/2]

```
quantity< volume , long double> boost::units::literals::operator"" _YL ( long double x )
```

Definition at line 99 of file literals.h.

5.4.1.1296 operator""_yL() [2/2]

```
quantity< volume , unsigned long long> boost::units::literals::operator"" _yL ( unsigned long long x )
```

Definition at line 99 of file literals.h.

5.4.1.1297 operator""_Ylm() [1/2]

```
quantity< luminous_flux , unsigned long long> boost::units::literals::operator"" _Ylm ( unsigned long long x )
```

Definition at line 88 of file literals.h.

5.4.1.1298 operator""_Ylm() [2/2]

```
quantity< luminous_flux , long double> boost::units::literals::operator"" _Ylm ( long double x )
```

Definition at line 88 of file literals.h.

5.4.1.1299 operator""_ylm() [1/2]

```
quantity< luminous_flux , unsigned long long> boost::units::literals::operator"" _ylm ( unsigned long long x )
```

Definition at line 88 of file literals.h.

5.4.1.1300 operator""_ylm() [2/2]

```
quantity< luminous_flux , long double> boost::units::literals::operator"" _ylm ( long double x )
```

Definition at line 88 of file literals.h.

5.4.1.1301 operator""_Ylx() [1/2]

```
quantity< illuminance , long double> boost::units::literals::operator"" _Ylx ( long double x )
```

Definition at line 89 of file literals.h.

5.4.1.1302 operator""_Ylx() [2/2]

```
quantity< illuminance , unsigned long long> boost::units::literals::operator"" _Ylx ( unsigned long long x )
```

Definition at line 89 of file literals.h.

5.4.1.1303 operator""_ylx() [1/2]

```
quantity< illuminance , long double> boost::units::literals::operator"" _ylx ( long double x )
```

Definition at line 89 of file literals.h.

5.4.1.1304 operator""_ylx() [2/2]

```
quantity< illuminance , unsigned long long> boost::units::literals::operator"" _ylx ( unsigned long long x )
```

Definition at line 89 of file literals.h.

5.4.1.1305 operator""_Ym() [1/2]

```
quantity< length , unsigned long long> boost::units::literals::operator"" _Ym ( unsigned long long x )
```

Definition at line 65 of file literals.h.

5.4.1.1306 operator""_ym() [1/2]

```
quantity< length , unsigned long long> boost::units::literals::operator"" _ym ( unsigned long long x )
```

Definition at line 65 of file literals.h.

5.4.1.1307 operator""_ym() [2/2]

```
quantity< length , long double> boost::units::literals::operator""_ym ( long double x )
```

Definition at line 65 of file literals.h.

5.4.1.1308 operator""_Ym() [2/2]

```
quantity< length , long double> boost::units::literals::operator""_Ym ( long double x )
```

Definition at line 65 of file literals.h.

5.4.1.1309 operator""_ymin() [1/2]

```
quantity< time , unsigned long long> boost::units::literals::operator""_ymin ( unsigned long long x )
```

Definition at line 94 of file literals.h.

5.4.1.1310 operator""_Ymin() [1/2]

```
quantity< time , long double> boost::units::literals::operator""_Ymin ( long double x )
```

Definition at line 94 of file literals.h.

5.4.1.1311 operator""_ymin() [2/2]

```
quantity< time , long double> boost::units::literals::operator""_ymin ( long double x )
```

Definition at line 94 of file literals.h.

5.4.1.1312 operator""_Ymin() [2/2]

```
quantity< time , unsigned long long> boost::units::literals::operator""_Ymin ( unsigned long long x )
```

Definition at line 94 of file literals.h.

5.4.1.1313 operator""_Ymol() [1/2]

```
quantity< amount , unsigned long long> boost::units::literals::operator""_Ymol ( unsigned long long x )
```

Definition at line 70 of file literals.h.

5.4.1.1314 operator""_ymol() [1/2]

```
quantity< amount , unsigned long long> boost::units::literals::operator""_ymol (  
    unsigned long long x )
```

Definition at line 70 of file literals.h.

5.4.1.1315 operator""_Ymol() [2/2]

```
quantity< amount , long double> boost::units::literals::operator""_Ymol (   
    long double x )
```

Definition at line 70 of file literals.h.

5.4.1.1316 operator""_ymol() [2/2]

```
quantity< amount , long double> boost::units::literals::operator""_ymol (   
    long double x )
```

Definition at line 70 of file literals.h.

5.4.1.1317 operator""_YN() [1/2]

```
quantity< force , long double> boost::units::literals::operator""_YN (   
    long double x )
```

Definition at line 75 of file literals.h.

5.4.1.1318 operator""_YN() [2/2]

```
quantity< force , unsigned long long> boost::units::literals::operator""_YN (   
    unsigned long long x )
```

Definition at line 75 of file literals.h.

5.4.1.1319 operator""_yN() [1/2]

```
quantity< force , unsigned long long> boost::units::literals::operator""_yN (   
    unsigned long long x )
```

Definition at line 75 of file literals.h.

5.4.1.1320 operator""_yN() [2/2]

```
quantity< force , long double> boost::units::literals::operator""_yN (   
    long double x )
```

Definition at line 75 of file literals.h.

5.4.1.1321 operator""_yohm() [1/2]

```
quantity< resistance , long double> boost::units::literals::operator"" _yohm (
    long double x )
```

Definition at line 82 of file literals.h.

5.4.1.1322 operator""_Yohm() [1/2]

```
quantity< resistance , long double> boost::units::literals::operator"" _Yohm (
    long double x )
```

Definition at line 82 of file literals.h.

5.4.1.1323 operator""_Yohm() [2/2]

```
quantity< resistance , unsigned long long> boost::units::literals::operator"" _Yohm (
    unsigned long long x )
```

Definition at line 82 of file literals.h.

5.4.1.1324 operator""_yohm() [2/2]

```
quantity< resistance , unsigned long long> boost::units::literals::operator"" _yohm (
    unsigned long long x )
```

Definition at line 82 of file literals.h.

5.4.1.1325 operator""_yPa() [1/2]

```
quantity< pressure , unsigned long long> boost::units::literals::operator"" _yPa (
    unsigned long long x )
```

Definition at line 76 of file literals.h.

5.4.1.1326 operator""_YPa() [1/2]

```
quantity< pressure , long double> boost::units::literals::operator"" _YPa (
    long double x )
```

Definition at line 76 of file literals.h.

5.4.1.1327 operator""_YPa() [2/2]

```
quantity< pressure , unsigned long long> boost::units::literals::operator"" _YPa (
    unsigned long long x )
```

Definition at line 76 of file literals.h.

5.4.1.1328 operator""_yPa() [2/2]

```
quantity< pressure , long double> boost::units::literals::operator""_yPa ( long double x )
```

Definition at line 76 of file literals.h.

5.4.1.1329 operator""_Yrad() [1/2]

```
quantity< plane_angle , unsigned long long> boost::units::literals::operator""_Yrad ( unsigned long long x )
```

Definition at line 73 of file literals.h.

5.4.1.1330 operator""_yrad() [1/2]

```
quantity< plane_angle , unsigned long long> boost::units::literals::operator""_yrad ( unsigned long long x )
```

Definition at line 73 of file literals.h.

5.4.1.1331 operator""_Yrad() [2/2]

```
quantity< plane_angle , long double> boost::units::literals::operator""_Yrad ( long double x )
```

Definition at line 73 of file literals.h.

5.4.1.1332 operator""_yrad() [2/2]

```
quantity< plane_angle , long double> boost::units::literals::operator""_yrad ( long double x )
```

Definition at line 73 of file literals.h.

5.4.1.1333 operator""_ys() [1/2]

```
quantity< time , unsigned long long> boost::units::literals::operator""_ys ( unsigned long long x )
```

Definition at line 67 of file literals.h.

5.4.1.1334 operator""_ys() [2/2]

```
quantity< time , long double> boost::units::literals::operator""_ys ( long double x )
```

Definition at line 67 of file literals.h.

5.4.1.1335 operator""_Ys() [1/2]

```
quantity< time , long double> boost::units::literals::operator""_Ys ( long double x )
```

Definition at line 67 of file literals.h.

5.4.1.1336 operator""_Ys() [2/2]

```
quantity< time , unsigned long long> boost::units::literals::operator""_Ys ( unsigned long long x )
```

Definition at line 67 of file literals.h.

5.4.1.1337 operator""_YS() [1/2]

```
quantity< conductance , unsigned long long> boost::units::literals::operator""_YS ( unsigned long long x )
```

Definition at line 83 of file literals.h.

5.4.1.1338 operator""_yS() [1/2]

```
quantity< conductance , unsigned long long> boost::units::literals::operator""_yS ( unsigned long long x )
```

Definition at line 83 of file literals.h.

5.4.1.1339 operator""_YS() [2/2]

```
quantity< conductance , long double> boost::units::literals::operator""_YS ( long double x )
```

Definition at line 83 of file literals.h.

5.4.1.1340 operator""_yS() [2/2]

```
quantity< conductance , long double> boost::units::literals::operator""_yS ( long double x )
```

Definition at line 83 of file literals.h.

5.4.1.1341 operator""_ysr() [1/2]

```
quantity< solid_angle , long double> boost::units::literals::operator""_ysr ( long double x )
```

Definition at line 74 of file literals.h.

5.4.1.1342 operator""_ysr() [2/2]

```
quantity< solid_angle , unsigned long long> boost::units::literals::operator""_ysr (  
    unsigned long long x )
```

Definition at line 74 of file literals.h.

5.4.1.1343 operator""_Ysr() [1/2]

```
quantity< solid_angle , unsigned long long> boost::units::literals::operator""_Ysr (   
    unsigned long long x )
```

Definition at line 74 of file literals.h.

5.4.1.1344 operator""_Ysr() [2/2]

```
quantity< solid_angle , long double> boost::units::literals::operator""_Ysr (   
    long double x )
```

Definition at line 74 of file literals.h.

5.4.1.1345 operator""_ySv() [1/2]

```
quantity< dose_equivalent , long double> boost::units::literals::operator""_ySv (   
    long double x )
```

Definition at line 92 of file literals.h.

5.4.1.1346 operator""_YSv() [1/2]

```
quantity< dose_equivalent , unsigned long long> boost::units::literals::operator""_YSv (   
    unsigned long long x )
```

Definition at line 92 of file literals.h.

5.4.1.1347 operator""_YSv() [2/2]

```
quantity< dose_equivalent , long double> boost::units::literals::operator""_YSv (   
    long double x )
```

Definition at line 92 of file literals.h.

5.4.1.1348 operator""_ySv() [2/2]

```
quantity< dose_equivalent , unsigned long long> boost::units::literals::operator""_ySv (   
    unsigned long long x )
```

Definition at line 92 of file literals.h.

5.4.1.1349 operator""_yT() [1/2]

```
quantity< magnetic_flux_density , unsigned long long> boost::units::literals::operator"" _yT ( unsigned long long x )
```

Definition at line 85 of file literals.h.

5.4.1.1350 operator""_yT() [2/2]

```
quantity< magnetic_flux_density , long double> boost::units::literals::operator"" _yT ( long double x )
```

Definition at line 85 of file literals.h.

5.4.1.1351 operator""_YT() [1/2]

```
quantity< magnetic_flux_density , unsigned long long> boost::units::literals::operator"" _YT ( unsigned long long x )
```

Definition at line 85 of file literals.h.

5.4.1.1352 operator""_YT() [2/2]

```
quantity< magnetic_flux_density , long double> boost::units::literals::operator"" _YT ( long double x )
```

Definition at line 85 of file literals.h.

5.4.1.1353 operator""_yt() [1/2]

```
quantity< mass , unsigned long long> boost::units::literals::operator"" _yt ( unsigned long long x )
```

Definition at line 100 of file literals.h.

5.4.1.1354 operator""_Yt() [1/2]

```
quantity< mass , long double> boost::units::literals::operator"" _Yt ( long double x )
```

Definition at line 100 of file literals.h.

5.4.1.1355 operator""_Yt() [2/2]

```
quantity< mass , unsigned long long> boost::units::literals::operator"" _Yt ( unsigned long long x )
```

Definition at line 100 of file literals.h.

5.4.1.1356 operator""_yt() [2/2]

```
quantity< mass , long double> boost::units::literals::operator""_yt ( long double x )
```

Definition at line 100 of file literals.h.

5.4.1.1357 operator""_YV() [1/2]

```
quantity< electric_potential , long double> boost::units::literals::operator""_YV ( long double x )
```

Definition at line 80 of file literals.h.

5.4.1.1358 operator""_YV() [2/2]

```
quantity< electric_potential , unsigned long long> boost::units::literals::operator""_YV ( unsigned long long x )
```

Definition at line 80 of file literals.h.

5.4.1.1359 operator""_yV() [1/2]

```
quantity< electric_potential , unsigned long long> boost::units::literals::operator""_yV ( unsigned long long x )
```

Definition at line 80 of file literals.h.

5.4.1.1360 operator""_yV() [2/2]

```
quantity< electric_potential , long double> boost::units::literals::operator""_yV ( long double x )
```

Definition at line 80 of file literals.h.

5.4.1.1361 operator""_yW() [1/2]

```
quantity< power , long double> boost::units::literals::operator""_yW ( long double x )
```

Definition at line 78 of file literals.h.

5.4.1.1362 operator""_yW() [2/2]

```
quantity< power , unsigned long long> boost::units::literals::operator""_yW ( unsigned long long x )
```

Definition at line 78 of file literals.h.

5.4.1.1363 operator""_YW() [1/2]

```
quantity< power , unsigned long long> boost::units::literals::operator""_YW (
    unsigned long long x )
```

Definition at line 78 of file literals.h.

5.4.1.1364 operator""_YW() [2/2]

```
quantity< power , long double> boost::units::literals::operator""_YW (
    long double x )
```

Definition at line 78 of file literals.h.

5.4.1.1365 operator""_YWB() [1/2]

```
quantity< magnetic_flux , unsigned long long> boost::units::literals::operator""_YWB (
    unsigned long long x )
```

Definition at line 84 of file literals.h.

5.4.1.1366 operator""_YWB() [2/2]

```
quantity< magnetic_flux , long double> boost::units::literals::operator""_YWB (
    long double x )
```

Definition at line 84 of file literals.h.

5.4.1.1367 operator""_yWB() [1/2]

```
quantity< magnetic_flux , unsigned long long> boost::units::literals::operator""_yWB (
    unsigned long long x )
```

Definition at line 84 of file literals.h.

5.4.1.1368 operator""_yWB() [2/2]

```
quantity< magnetic_flux , long double> boost::units::literals::operator""_yWB (
    long double x )
```

Definition at line 84 of file literals.h.

5.4.1.1369 operator""_ZA() [1/2]

```
quantity< current , unsigned long long> boost::units::literals::operator""_ZA (
    unsigned long long x )
```

Definition at line 68 of file literals.h.

5.4.1.1370 operator""_zA() [1/2]

```
quantity< current , unsigned long long> boost::units::literals::operator""_zA (  
    unsigned long long x )
```

Definition at line 68 of file literals.h.

5.4.1.1371 operator""_ZA() [2/2]

```
quantity< current , long double> boost::units::literals::operator""_ZA (   
    long double x )
```

Definition at line 68 of file literals.h.

5.4.1.1372 operator""_zA() [2/2]

```
quantity< current , long double> boost::units::literals::operator""_zA (   
    long double x )
```

Definition at line 68 of file literals.h.

5.4.1.1373 operator""_zBq() [1/2]

```
quantity< activity , long double> boost::units::literals::operator""_zBq (   
    long double x )
```

Definition at line 90 of file literals.h.

5.4.1.1374 operator""_ZBq() [1/2]

```
quantity< activity , unsigned long long> boost::units::literals::operator""_ZBq (   
    unsigned long long x )
```

Definition at line 90 of file literals.h.

5.4.1.1375 operator""_zBq() [2/2]

```
quantity< activity , unsigned long long> boost::units::literals::operator""_zBq (   
    unsigned long long x )
```

Definition at line 90 of file literals.h.

5.4.1.1376 operator""_ZBq() [2/2]

```
quantity< activity , long double> boost::units::literals::operator""_ZBq (   
    long double x )
```

Definition at line 90 of file literals.h.

5.4.1.1377 operator""_ZC() [1/2]

```
quantity< electric_charge , unsigned long long> boost::units::literals::operator""_ZC( unsigned long long x )
```

Definition at line 79 of file literals.h.

5.4.1.1378 operator""_zC() [1/2]

```
quantity< electric_charge , unsigned long long> boost::units::literals::operator""_zC( unsigned long long x )
```

Definition at line 79 of file literals.h.

5.4.1.1379 operator""_ZC() [2/2]

```
quantity< electric_charge , long double> boost::units::literals::operator""_ZC( long double x )
```

Definition at line 79 of file literals.h.

5.4.1.1380 operator""_zC() [2/2]

```
quantity< electric_charge , long double> boost::units::literals::operator""_zC( long double x )
```

Definition at line 79 of file literals.h.

5.4.1.1381 operator""_zcd() [1/2]

```
quantity< luminous_intensity , unsigned long long> boost::units::literals::operator""_zcd( unsigned long long x )
```

Definition at line 71 of file literals.h.

5.4.1.1382 operator""_Zcd() [1/2]

```
quantity< luminous_intensity , unsigned long long> boost::units::literals::operator""_Zcd( unsigned long long x )
```

Definition at line 71 of file literals.h.

5.4.1.1383 operator""_zcd() [2/2]

```
quantity< luminous_intensity , long double> boost::units::literals::operator""_zcd( long double x )
```

Definition at line 71 of file literals.h.

5.4.1.1384 operator""_Zcd() [2/2]

```
quantity< luminous_intensity , long double> boost::units::literals::operator""_Zcd ( long double x )
```

Definition at line 71 of file literals.h.

5.4.1.1385 operator""_Zday() [1/2]

```
quantity< time , unsigned long long> boost::units::literals::operator""_Zday ( unsigned long long x )
```

Definition at line 96 of file literals.h.

5.4.1.1386 operator""_zday() [1/2]

```
quantity< time , unsigned long long> boost::units::literals::operator""_zday ( unsigned long long x )
```

Definition at line 96 of file literals.h.

5.4.1.1387 operator""_zday() [2/2]

```
quantity< time , long double> boost::units::literals::operator""_zday ( long double x )
```

Definition at line 96 of file literals.h.

5.4.1.1388 operator""_Zday() [2/2]

```
quantity< time , long double> boost::units::literals::operator""_Zday ( long double x )
```

Definition at line 96 of file literals.h.

5.4.1.1389 operator""_zdeg() [1/2]

```
quantity< plane_angle , unsigned long long> boost::units::literals::operator""_zdeg ( unsigned long long x )
```

Definition at line 97 of file literals.h.

5.4.1.1390 operator""_Zdeg() [1/2]

```
quantity< plane_angle , unsigned long long> boost::units::literals::operator""_Zdeg ( unsigned long long x )
```

Definition at line 97 of file literals.h.

5.4.1.1391 operator""_zdeg() [2/2]

```
quantity< plane_angle , long double> boost::units::literals::operator"" _zdeg ( long double x )
```

Definition at line 97 of file literals.h.

5.4.1.1392 operator""_Zdeg() [2/2]

```
quantity< plane_angle , long double> boost::units::literals::operator"" _Zdeg ( long double x )
```

Definition at line 97 of file literals.h.

5.4.1.1393 operator""_ZdegC() [1/2]

```
quantity< temperature , long double> boost::units::literals::operator"" _ZdegC ( long double x )
```

Definition at line 87 of file literals.h.

5.4.1.1394 operator""_zdegC() [1/2]

```
quantity< temperature , long double> boost::units::literals::operator"" _zdegC ( long double x )
```

Definition at line 87 of file literals.h.

5.4.1.1395 operator""_ZdegC() [2/2]

```
quantity< temperature , unsigned long long> boost::units::literals::operator"" _ZdegC ( unsigned long long x )
```

Definition at line 87 of file literals.h.

5.4.1.1396 operator""_zdegC() [2/2]

```
quantity< temperature , unsigned long long> boost::units::literals::operator"" _zdegC ( unsigned long long x )
```

Definition at line 87 of file literals.h.

5.4.1.1397 operator""_zF() [1/2]

```
quantity< capacitance , long double> boost::units::literals::operator"" _zF ( long double x )
```

Definition at line 81 of file literals.h.

5.4.1.1398 operator""_ZF() [1/2]

```
quantity< capacitance , unsigned long long> boost::units::literals::operator""_ZF( unsigned long long x )
```

Definition at line 81 of file literals.h.

5.4.1.1399 operator""_zF() [2/2]

```
quantity< capacitance , unsigned long long> boost::units::literals::operator""_zF( unsigned long long x )
```

Definition at line 81 of file literals.h.

5.4.1.1400 operator""_ZF() [2/2]

```
quantity< capacitance , long double> boost::units::literals::operator""_ZF( long double x )
```

Definition at line 81 of file literals.h.

5.4.1.1401 operator""_zg() [1/2]

```
quantity< mass , long double> boost::units::literals::operator""_zg( long double x )
```

Definition at line 66 of file literals.h.

5.4.1.1402 operator""_Zg() [1/2]

```
quantity< mass , unsigned long long> boost::units::literals::operator""_Zg( unsigned long long x )
```

Definition at line 66 of file literals.h.

5.4.1.1403 operator""_Zg() [2/2]

```
quantity< mass , long double> boost::units::literals::operator""_Zg( long double x )
```

Definition at line 66 of file literals.h.

5.4.1.1404 operator""_zg() [2/2]

```
quantity< mass , unsigned long long> boost::units::literals::operator""_zg( unsigned long long x )
```

Definition at line 66 of file literals.h.

5.4.1.1405 operator""_ZGy() [1/2]

```
quantity< absorbed_dose , long double> boost::units::literals::operator""_ZGy ( long double x )
```

Definition at line 91 of file literals.h.

5.4.1.1406 operator""_ZGy() [2/2]

```
quantity< absorbed_dose , unsigned long long> boost::units::literals::operator""_ZGy ( unsigned long long x )
```

Definition at line 91 of file literals.h.

5.4.1.1407 operator""_zGy() [1/2]

```
quantity< absorbed_dose , long double> boost::units::literals::operator""_zGy ( long double x )
```

Definition at line 91 of file literals.h.

5.4.1.1408 operator""_zGy() [2/2]

```
quantity< absorbed_dose , unsigned long long> boost::units::literals::operator""_zGy ( unsigned long long x )
```

Definition at line 91 of file literals.h.

5.4.1.1409 operator""_zH() [1/2]

```
quantity< inductance , long double> boost::units::literals::operator""_zH ( long double x )
```

Definition at line 86 of file literals.h.

5.4.1.1410 operator""_zH() [1/2]

```
quantity< inductance , long double> boost::units::literals::operator""_zH ( long double x )
```

Definition at line 86 of file literals.h.

5.4.1.1411 operator""_zH() [2/2]

```
quantity< inductance , unsigned long long> boost::units::literals::operator""_zH ( unsigned long long x )
```

Definition at line 86 of file literals.h.

5.4.1.1412 operator""_zH() [2/2]

```
quantity< inductance , unsigned long long> boost::units::literals::operator""_zH (  
    unsigned long long x )
```

Definition at line 86 of file literals.h.

5.4.1.1413 operator""_zh() [1/2]

```
quantity< time , unsigned long long> boost::units::literals::operator""_zh (   
    unsigned long long x )
```

Definition at line 95 of file literals.h.

5.4.1.1414 operator""_Zh() [1/2]

```
quantity< time , long double> boost::units::literals::operator""_Zh (   
    long double x )
```

Definition at line 95 of file literals.h.

5.4.1.1415 operator""_zh() [2/2]

```
quantity< time , long double> boost::units::literals::operator""_zh (   
    long double x )
```

Definition at line 95 of file literals.h.

5.4.1.1416 operator""_Zh() [2/2]

```
quantity< time , unsigned long long> boost::units::literals::operator""_Zh (   
    unsigned long long x )
```

Definition at line 95 of file literals.h.

5.4.1.1417 operator""_zHz() [1/2]

```
quantity< frequency , unsigned long long> boost::units::literals::operator""_zHz (   
    unsigned long long x )
```

Definition at line 72 of file literals.h.

5.4.1.1418 operator""_zHz() [2/2]

```
quantity< frequency , long double> boost::units::literals::operator""_zHz (   
    long double x )
```

Definition at line 72 of file literals.h.

5.4.1.1419 operator""_ZHz() [1/2]

```
quantity< frequency , unsigned long long> boost::units::literals::operator"" _ZHz (  
    unsigned long long x )
```

Definition at line 72 of file literals.h.

5.4.1.1420 operator""_ZHz() [2/2]

```
quantity< frequency , long double> boost::units::literals::operator"" _ZHz (   
    long double x )
```

Definition at line 72 of file literals.h.

5.4.1.1421 operator""_zJ() [1/2]

```
quantity< energy , long double> boost::units::literals::operator"" _zJ (   
    long double x )
```

Definition at line 77 of file literals.h.

5.4.1.1422 operator""_zJ() [1/2]

```
quantity< energy , long double> boost::units::literals::operator"" _zJ (   
    long double x )
```

Definition at line 77 of file literals.h.

5.4.1.1423 operator""_zJ() [2/2]

```
quantity< energy , unsigned long long> boost::units::literals::operator"" _zJ (   
    unsigned long long x )
```

Definition at line 77 of file literals.h.

5.4.1.1424 operator""_zJ() [2/2]

```
quantity< energy , unsigned long long> boost::units::literals::operator"" _zJ (   
    unsigned long long x )
```

Definition at line 77 of file literals.h.

5.4.1.1425 operator""_zK() [1/2]

```
quantity< temperature , unsigned long long> boost::units::literals::operator"" _zK (   
    unsigned long long x )
```

Definition at line 69 of file literals.h.

5.4.1.1426 operator""_ZK() [1/2]

```
quantity< temperature , unsigned long long> boost::units::literals::operator""_ZK (  
    unsigned long long x )
```

Definition at line 69 of file literals.h.

5.4.1.1427 operator""_zK() [2/2]

```
quantity< temperature , long double> boost::units::literals::operator""_zK (   
    long double x )
```

Definition at line 69 of file literals.h.

5.4.1.1428 operator""_ZK() [2/2]

```
quantity< temperature , long double> boost::units::literals::operator""_ZK (   
    long double x )
```

Definition at line 69 of file literals.h.

5.4.1.1429 operator""_zkat() [1/2]

```
quantity< catalytic_activity , unsigned long long> boost::units::literals::operator""_zkat (   
    unsigned long long x )
```

Definition at line 93 of file literals.h.

5.4.1.1430 operator""_zkat() [2/2]

```
quantity< catalytic_activity , long double> boost::units::literals::operator""_zkat (   
    long double x )
```

Definition at line 93 of file literals.h.

5.4.1.1431 operator""_Zkat() [1/2]

```
quantity< catalytic_activity , unsigned long long> boost::units::literals::operator""_Zkat (   
    unsigned long long x )
```

Definition at line 93 of file literals.h.

5.4.1.1432 operator""_Zkat() [2/2]

```
quantity< catalytic_activity , long double> boost::units::literals::operator""_Zkat (   
    long double x )
```

Definition at line 93 of file literals.h.

5.4.1.1433 operator""_Zl() [1/2]

```
quantity< volume , long double> boost::units::literals::operator""_Zl ( long double x )
```

Definition at line 98 of file literals.h.

5.4.1.1434 operator""_zl() [1/2]

```
quantity< volume , unsigned long long> boost::units::literals::operator""_zl ( unsigned long long x )
```

Definition at line 98 of file literals.h.

5.4.1.1435 operator""_Zl() [2/2]

```
quantity< volume , unsigned long long> boost::units::literals::operator""_Zl ( unsigned long long x )
```

Definition at line 98 of file literals.h.

5.4.1.1436 operator""_zl() [2/2]

```
quantity< volume , long double> boost::units::literals::operator""_zl ( long double x )
```

Definition at line 98 of file literals.h.

5.4.1.1437 operator""_zL() [1/2]

```
quantity< volume , long double> boost::units::literals::operator""_zL ( long double x )
```

Definition at line 99 of file literals.h.

5.4.1.1438 operator""_zL() [1/2]

```
quantity< volume , unsigned long long> boost::units::literals::operator""_zL ( unsigned long long x )
```

Definition at line 99 of file literals.h.

5.4.1.1439 operator""_zL() [2/2]

```
quantity< volume , unsigned long long> boost::units::literals::operator""_zL ( unsigned long long x )
```

Definition at line 99 of file literals.h.

5.4.1.1440 operator""_ZL() [2/2]

```
quantity< volume , long double> boost::units::literals::operator""_ZL ( long double x )
```

Definition at line 99 of file literals.h.

5.4.1.1441 operator""_Zlm() [1/2]

```
quantity< luminous_flux , long double> boost::units::literals::operator""_Zlm ( long double x )
```

Definition at line 88 of file literals.h.

5.4.1.1442 operator""_Zlm() [2/2]

```
quantity< luminous_flux , unsigned long long> boost::units::literals::operator""_Zlm ( unsigned long long x )
```

Definition at line 88 of file literals.h.

5.4.1.1443 operator""_zlm() [1/2]

```
quantity< luminous_flux , unsigned long long> boost::units::literals::operator""_zlm ( unsigned long long x )
```

Definition at line 88 of file literals.h.

5.4.1.1444 operator""_zlm() [2/2]

```
quantity< luminous_flux , long double> boost::units::literals::operator""_zlm ( long double x )
```

Definition at line 88 of file literals.h.

5.4.1.1445 operator""_Zlx() [1/2]

```
quantity< illuminance , unsigned long long> boost::units::literals::operator""_Zlx ( unsigned long long x )
```

Definition at line 89 of file literals.h.

5.4.1.1446 operator""_zlx() [1/2]

```
quantity< illuminance , unsigned long long> boost::units::literals::operator""_zlx ( unsigned long long x )
```

Definition at line 89 of file literals.h.

5.4.1.1447 operator""_Zlx() [2/2]

```
quantity< illuminance , long double> boost::units::literals::operator""_Zlx ( long double x )
```

Definition at line 89 of file literals.h.

5.4.1.1448 operator""_zlx() [2/2]

```
quantity< illuminance , long double> boost::units::literals::operator""_zlx ( long double x )
```

Definition at line 89 of file literals.h.

5.4.1.1449 operator""_Zm() [1/2]

```
quantity< length , unsigned long long> boost::units::literals::operator""_Zm ( unsigned long long x )
```

Definition at line 65 of file literals.h.

5.4.1.1450 operator""_zm() [1/2]

```
quantity< length , unsigned long long> boost::units::literals::operator""_zm ( unsigned long long x )
```

Definition at line 65 of file literals.h.

5.4.1.1451 operator""_zm() [2/2]

```
quantity< length , long double> boost::units::literals::operator""_zm ( long double x )
```

Definition at line 65 of file literals.h.

5.4.1.1452 operator""_Zm() [2/2]

```
quantity< length , long double> boost::units::literals::operator""_Zm ( long double x )
```

Definition at line 65 of file literals.h.

5.4.1.1453 operator""_Zmin() [1/2]

```
quantity< time , unsigned long long> boost::units::literals::operator""_Zmin ( unsigned long long x )
```

Definition at line 94 of file literals.h.

5.4.1.1454 operator""_Zmin() [2/2]

```
quantity< time , long double> boost::units::literals::operator""_Zmin (  
    long double x )
```

Definition at line 94 of file literals.h.

5.4.1.1455 operator""_zmin() [1/2]

```
quantity< time , long double> boost::units::literals::operator""_zmin (   
    long double x )
```

Definition at line 94 of file literals.h.

5.4.1.1456 operator""_zmin() [2/2]

```
quantity< time , unsigned long long> boost::units::literals::operator""_zmin (   
    unsigned long long x )
```

Definition at line 94 of file literals.h.

5.4.1.1457 operator""_zmol() [1/2]

```
quantity< amount , long double> boost::units::literals::operator""_zmol (   
    long double x )
```

Definition at line 70 of file literals.h.

5.4.1.1458 operator""_Zmol() [1/2]

```
quantity< amount , long double> boost::units::literals::operator""_Zmol (   
    long double x )
```

Definition at line 70 of file literals.h.

5.4.1.1459 operator""_zmol() [2/2]

```
quantity< amount , unsigned long long> boost::units::literals::operator""_zmol (   
    unsigned long long x )
```

Definition at line 70 of file literals.h.

5.4.1.1460 operator""_Zmol() [2/2]

```
quantity< amount , unsigned long long> boost::units::literals::operator""_Zmol (   
    unsigned long long x )
```

Definition at line 70 of file literals.h.

5.4.1.1461 operator""_ZN() [1/2]

```
quantity< force , long double> boost::units::literals::operator"" _ZN ( long double x )
```

Definition at line 75 of file literals.h.

5.4.1.1462 operator""_ZN() [2/2]

```
quantity< force , unsigned long long> boost::units::literals::operator"" _ZN ( unsigned long long x )
```

Definition at line 75 of file literals.h.

5.4.1.1463 operator""_ZN() [1/2]

```
quantity< force , unsigned long long> boost::units::literals::operator"" _ZN ( unsigned long long x )
```

Definition at line 75 of file literals.h.

5.4.1.1464 operator""_ZN() [2/2]

```
quantity< force , long double> boost::units::literals::operator"" _ZN ( long double x )
```

Definition at line 75 of file literals.h.

5.4.1.1465 operator""_zohm() [1/2]

```
quantity< resistance , long double> boost::units::literals::operator"" _zohm ( long double x )
```

Definition at line 82 of file literals.h.

5.4.1.1466 operator""_Zohm() [1/2]

```
quantity< resistance , long double> boost::units::literals::operator"" _Zohm ( long double x )
```

Definition at line 82 of file literals.h.

5.4.1.1467 operator""_zohm() [2/2]

```
quantity< resistance , unsigned long long> boost::units::literals::operator"" _zohm ( unsigned long long x )
```

Definition at line 82 of file literals.h.

5.4.1.1468 operator""_Zohm() [2/2]

```
quantity< resistance , unsigned long long> boost::units::literals::operator""_Zohm (  
    unsigned long long x )
```

Definition at line 82 of file literals.h.

5.4.1.1469 operator""_ZPa() [1/2]

```
quantity< pressure , unsigned long long> boost::units::literals::operator""_ZPa (   
    unsigned long long x )
```

Definition at line 76 of file literals.h.

5.4.1.1470 operator""_zPa() [1/2]

```
quantity< pressure , long double> boost::units::literals::operator""_zPa (   
    long double x )
```

Definition at line 76 of file literals.h.

5.4.1.1471 operator""_ZPa() [2/2]

```
quantity< pressure , long double> boost::units::literals::operator""_ZPa (   
    long double x )
```

Definition at line 76 of file literals.h.

5.4.1.1472 operator""_zPa() [2/2]

```
quantity< pressure , unsigned long long> boost::units::literals::operator""_zPa (   
    unsigned long long x )
```

Definition at line 76 of file literals.h.

5.4.1.1473 operator""_zrad() [1/2]

```
quantity< plane_angle , unsigned long long> boost::units::literals::operator""_zrad (   
    unsigned long long x )
```

Definition at line 73 of file literals.h.

5.4.1.1474 operator""_Zrad() [1/2]

```
quantity< plane_angle , long double> boost::units::literals::operator""_Zrad (   
    long double x )
```

Definition at line 73 of file literals.h.

5.4.1.1475 operator""_Zrad() [2/2]

```
quantity< plane_angle , unsigned long long> boost::units::literals::operator""_Zrad ( unsigned long long x )
```

Definition at line 73 of file literals.h.

5.4.1.1476 operator""_zrad() [2/2]

```
quantity< plane_angle , long double> boost::units::literals::operator""_zrad ( long double x )
```

Definition at line 73 of file literals.h.

5.4.1.1477 operator""_zs() [1/2]

```
quantity< time , long double> boost::units::literals::operator""_zs ( long double x )
```

Definition at line 67 of file literals.h.

5.4.1.1478 operator""_Zs() [1/2]

```
quantity< time , unsigned long long> boost::units::literals::operator""_Zs ( unsigned long long x )
```

Definition at line 67 of file literals.h.

5.4.1.1479 operator""_Zs() [2/2]

```
quantity< time , long double> boost::units::literals::operator""_Zs ( long double x )
```

Definition at line 67 of file literals.h.

5.4.1.1480 operator""_zs() [2/2]

```
quantity< time , unsigned long long> boost::units::literals::operator""_zs ( unsigned long long x )
```

Definition at line 67 of file literals.h.

5.4.1.1481 operator""_ZS() [1/2]

```
quantity< conductance , unsigned long long> boost::units::literals::operator""_ZS ( unsigned long long x )
```

Definition at line 83 of file literals.h.

5.4.1.1482 operator""_ZS() [2/2]

```
quantity< conductance , long double> boost::units::literals::operator""_ZS ( long double x )
```

Definition at line 83 of file literals.h.

5.4.1.1483 operator""_zS() [1/2]

```
quantity< conductance , long double> boost::units::literals::operator""_zS ( long double x )
```

Definition at line 83 of file literals.h.

5.4.1.1484 operator""_zs() [2/2]

```
quantity< conductance , unsigned long long> boost::units::literals::operator""_zs ( unsigned long long x )
```

Definition at line 83 of file literals.h.

5.4.1.1485 operator""_zsr() [1/2]

```
quantity< solid_angle , long double> boost::units::literals::operator""_zsr ( long double x )
```

Definition at line 74 of file literals.h.

5.4.1.1486 operator""_zs() [2/2]

```
quantity< solid_angle , unsigned long long> boost::units::literals::operator""_zs ( unsigned long long x )
```

Definition at line 74 of file literals.h.

5.4.1.1487 operator""_Zsr() [1/2]

```
quantity< solid_angle , unsigned long long> boost::units::literals::operator""_Zsr ( unsigned long long x )
```

Definition at line 74 of file literals.h.

5.4.1.1488 operator""_Zs() [2/2]

```
quantity< solid_angle , long double> boost::units::literals::operator""_Zs ( long double x )
```

Definition at line 74 of file literals.h.

5.4.1.1489 operator""_ZSv() [1/2]

```
quantity< dose_equivalent , long double> boost::units::literals::operator""_ZSv ( long double x )
```

Definition at line 92 of file literals.h.

5.4.1.1490 operator""_zSv() [1/2]

```
quantity< dose_equivalent , unsigned long long> boost::units::literals::operator""_zSv ( unsigned long long x )
```

Definition at line 92 of file literals.h.

5.4.1.1491 operator""_ZSv() [2/2]

```
quantity< dose_equivalent , unsigned long long> boost::units::literals::operator""_ZSv ( unsigned long long x )
```

Definition at line 92 of file literals.h.

5.4.1.1492 operator""_zSv() [2/2]

```
quantity< dose_equivalent , long double> boost::units::literals::operator""_zSv ( long double x )
```

Definition at line 92 of file literals.h.

5.4.1.1493 operator""_zT() [1/2]

```
quantity< magnetic_flux_density , long double> boost::units::literals::operator""_zT ( long double x )
```

Definition at line 85 of file literals.h.

5.4.1.1494 operator""_ZT() [1/2]

```
quantity< magnetic_flux_density , unsigned long long> boost::units::literals::operator""_ZT ( unsigned long long x )
```

Definition at line 85 of file literals.h.

5.4.1.1495 operator""_zT() [2/2]

```
quantity< magnetic_flux_density , unsigned long long> boost::units::literals::operator""_zT ( unsigned long long x )
```

Definition at line 85 of file literals.h.

5.4.1.1496 operator""_ZT() [2/2]

```
quantity< magnetic_flux_density , long double> boost::units::literals::operator""_ZT ( long double x )
```

Definition at line 85 of file literals.h.

5.4.1.1497 operator""_Zt() [1/2]

```
quantity< mass , long double> boost::units::literals::operator""_Zt ( long double x )
```

Definition at line 100 of file literals.h.

5.4.1.1498 operator""_Zt() [2/2]

```
quantity< mass , unsigned long long> boost::units::literals::operator""_Zt ( unsigned long long x )
```

Definition at line 100 of file literals.h.

5.4.1.1499 operator""_zt() [1/2]

```
quantity< mass , long double> boost::units::literals::operator""_zt ( long double x )
```

Definition at line 100 of file literals.h.

5.4.1.1500 operator""_zt() [2/2]

```
quantity< mass , unsigned long long> boost::units::literals::operator""_zt ( unsigned long long x )
```

Definition at line 100 of file literals.h.

5.4.1.1501 operator""_ZV() [1/2]

```
quantity< electric_potential , unsigned long long> boost::units::literals::operator""_ZV ( unsigned long long x )
```

Definition at line 80 of file literals.h.

5.4.1.1502 operator""_zV() [1/2]

```
quantity< electric_potential , unsigned long long> boost::units::literals::operator""_zV ( unsigned long long x )
```

Definition at line 80 of file literals.h.

5.4.1.1503 operator""_ZV() [2/2]

```
quantity< electric_potential , long double> boost::units::literals::operator""_ZV(
    long double x )
```

Definition at line 80 of file literals.h.

5.4.1.1504 operator""_zV() [2/2]

```
quantity< electric_potential , long double> boost::units::literals::operator""_zV(
    long double x )
```

Definition at line 80 of file literals.h.

5.4.1.1505 operator""_zW() [1/2]

```
quantity< power , unsigned long long> boost::units::literals::operator""_zW(
    unsigned long long x )
```

Definition at line 78 of file literals.h.

5.4.1.1506 operator""_ZW() [1/2]

```
quantity< power , long double> boost::units::literals::operator""_ZW(
    long double x )
```

Definition at line 78 of file literals.h.

5.4.1.1507 operator""_zW() [2/2]

```
quantity< power , long double> boost::units::literals::operator""_zW(
    long double x )
```

Definition at line 78 of file literals.h.

5.4.1.1508 operator""_ZW() [2/2]

```
quantity< power , unsigned long long> boost::units::literals::operator""_ZW(
    unsigned long long x )
```

Definition at line 78 of file literals.h.

5.4.1.1509 operator""_zWb() [1/2]

```
quantity< magnetic_flux , unsigned long long> boost::units::literals::operator""_zWb(
    unsigned long long x )
```

Definition at line 84 of file literals.h.

5.4.1.1510 operator""_zWb() [2/2]

```
quantity< magnetic_flux , long double> boost::units::literals::operator"" _zWb ( long double x )
```

Definition at line 84 of file literals.h.

5.4.1.1511 operator""_ZWb() [1/2]

```
quantity< magnetic_flux , long double> boost::units::literals::operator"" _ZWb ( long double x )
```

Definition at line 84 of file literals.h.

5.4.1.1512 operator""_ZWb() [2/2]

```
quantity< magnetic_flux , unsigned long long> boost::units::literals::operator"" _ZWb ( unsigned long long x )
```

Definition at line 84 of file literals.h.

5.5 oCpt Namespace Reference

Namespaces

- [components](#)
- [protocol](#)
- [vessels](#)

Classes

- class [Actuator](#)
- class [ActuatorTask](#)
- class [ARM](#)
- class [Boatswain](#)
- class [Captain](#)
- class [CommunicationTask](#)
- class [CoveragePathTask](#)

An object representing a coverage path task.

- class [DredgeTask](#)
- An Object representing a dredging task.*
- class [FollowTask](#)
- An object representing a follow the target task.*
- class [iActuator](#)
- class [iBoatswain](#)
- class [iCaptain](#)
- class [iComm](#)
- class [iController](#)

- class [iSensor](#)
- class [iTask](#)

Task interface, all tasks need to adhere to this structure.
- class [iVessel](#)
- class [LogTask](#)

An Object representing a data logging task.
- class [LoRa](#)
- class [oCptException](#)
- class [PathTask](#)

An object representing a normal A to B type of path planning.
- class [RouteTask](#)
- class [Sensor](#)
- class [SensorTask](#)
- class [Task](#)
- class [Vessel](#)
- class [WorkTask](#)
- class [World](#)

5.6 oCpt::components Namespace Reference

Namespaces

- [comm](#)
- [controller](#)
- [sensors](#)

5.7 oCpt::components::comm Namespace Reference

Classes

- class [LoRa_RN2483](#)

5.8 oCpt::components::controller Namespace Reference

Classes

- class [BBB](#)

5.9 oCpt::components::sensors Namespace Reference

Classes

- class [ControlKalmanIMU](#)
- class [Gps](#)
- class [KalmanIMU](#)
- class [OrientationMeasurementKalmanIMU](#)
- class [OrientationMeasurementModelKalmanIMU](#)
- class [PositionMeasurementKalmanIMU](#)
- class [PositionMeasurementModelKalmanIMU](#)
- class [PT100](#)
- class [Razor](#)
- class [StateKalmanIMU](#)
- class [SystemModelKalmanIMU](#)

5.10 oCpt::protocol Namespace Reference

Classes

- class [adc](#)
- class [gpio](#)
- class [Serial](#)
- class [userspace](#)

5.11 oCpt::vessels Namespace Reference

Classes

- class [Meetcatamaran](#)

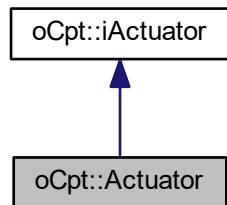
Chapter 6

Class Documentation

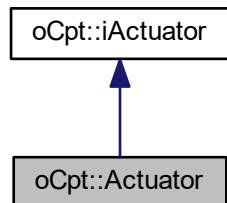
6.1 oCpt::Actuator Class Reference

```
#include <Actuator.h>
```

Inheritance diagram for oCpt::Actuator:



Collaboration diagram for oCpt::Actuator:



Public Member Functions

- [Actuator \(\)](#)
- virtual [~Actuator \(\)](#) override
- virtual void [setActuator \(\)](#) override
- virtual void [run \(\)](#) override
- virtual void [stop \(\)](#) override

Additional Inherited Members

6.1.1 Detailed Description

Definition at line 32 of file Actuator.h.

6.1.2 Constructor & Destructor Documentation

6.1.2.1 [Actuator\(\)](#)

```
oCpt::Actuator::Actuator ( )
```

Definition at line 17 of file Actuator.cpp.

6.1.2.2 [~Actuator\(\)](#)

```
oCpt::Actuator::~Actuator ( ) [override], [virtual]
```

Definition at line 21 of file Actuator.cpp.

6.1.3 Member Function Documentation

6.1.3.1 [run\(\)](#)

```
void oCpt::Actuator::run ( ) [override], [virtual]
```

Implements [oCpt::iActuator](#).

Definition at line 29 of file Actuator.cpp.

6.1.3.2 [setActuator\(\)](#)

```
void oCpt::Actuator::setActuator ( ) [override], [virtual]
```

Implements [oCpt::iActuator](#).

Definition at line 25 of file Actuator.cpp.

6.1.3.3 stop()

```
void oCpt::Actuator::stop ( ) [override], [virtual]
```

Implements [oCpt::iActuator](#).

Definition at line 33 of file Actuator.cpp.

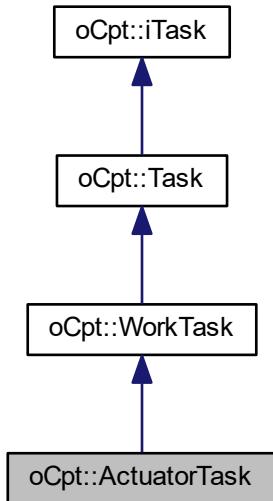
The documentation for this class was generated from the following files:

- /projects/mti/ohCaptain/ohCaptain/include/Core/[Actuator.h](#)
- /projects/mti/ohCaptain/ohCaptain/src/Core/[Actuator.cpp](#)

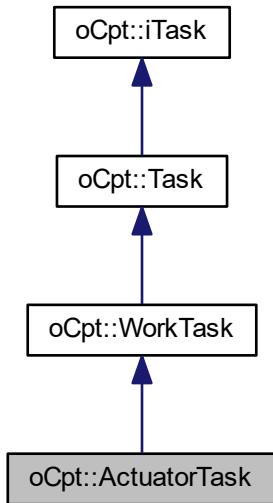
6.2 oCpt::ActuatorTask Class Reference

```
#include <Task.h>
```

Inheritance diagram for oCpt::ActuatorTask:



Collaboration diagram for oCpt::ActuatorTask:



Public Member Functions

- [ActuatorTask \(Vessel::ptr vessel, bool concurrent=true\)](#)
- virtual [~ActuatorTask \(\)](#)

Additional Inherited Members

6.2.1 Detailed Description

Definition at line 312 of file Task.h.

6.2.2 Constructor & Destructor Documentation

6.2.2.1 ActuatorTask()

```
oCpt::ActuatorTask::ActuatorTask (
    Vessel::ptr vessel,
    bool concurrent = true )
```

Definition at line 77 of file Task.cpp.

6.2.2.2 ~ActuatorTask()

```
oCpt::ActuatorTask::~ActuatorTask ( ) [virtual]
```

Definition at line 79 of file Task.cpp.

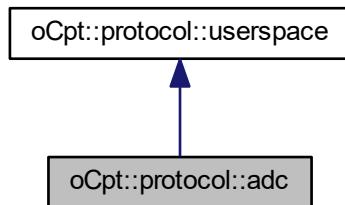
The documentation for this class was generated from the following files:

- /projects/mti/ohCaptain/ohCaptain/include/Core/[Task.h](#)
- /projects/mti/ohCaptain/ohCaptain/src/Core/[Task.cpp](#)

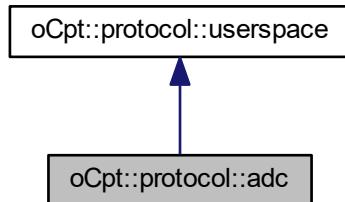
6.3 oCpt::protocol::adc Class Reference

```
#include <Controller.h>
```

Inheritance diagram for oCpt::protocol::adc:



Collaboration diagram for oCpt::protocol::adc:



Public Types

- `typedef boost::shared_ptr< adc > ptr`

Public Member Functions

- `adc` (`uint8_t id, uint8_t device, std::string modName=""`)
- virtual `~adc` ()
- `uint16_t & getValue` ()
- `bool operator==` (`const adc &rhs`)
- `bool compare` (`const uint8_t &id, const uint8_t &device=0`)

Private Attributes

- `uint8_t id_ = 0`
- `uint8_t device_ = 0`
- `std::string path_ = ""`
- `uint16_t value_ = 0`

Additional Inherited Members

6.3.1 Detailed Description

The Analogue to Digital converter class. This class reads the voltage of an analogue pin, from user space.

Definition at line 89 of file Controller.h.

6.3.2 Member Typedef Documentation

6.3.2.1 ptr

```
typedef boost::shared_ptr<adc> oCpt::protocol::adc::ptr
```

Definition at line 91 of file Controller.h.

6.3.3 Constructor & Destructor Documentation

6.3.3.1 adc()

```
oCpt::protocol::adc::adc (
    uint8_t id,
    uint8_t device,
    std::string modName = "" )
```

The constructor of the adc class

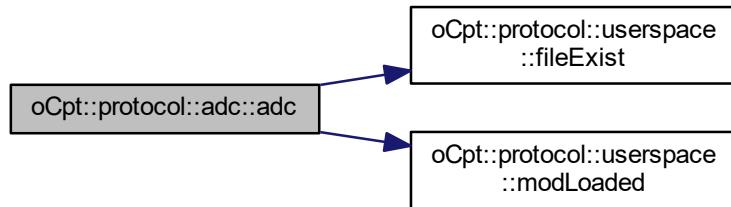
Parameters

<code>id</code>	the pin ID as an <code>uint8_t</code> value
<code>device</code>	the device or chip which handles the communication with the analogue pins
<code>modName</code>	the name of the modules which needs to be loaded TODO check if it is always needed to load a module

Definition at line 61 of file Controller.cpp.

References ADC_IO_BASE_PATH, ADC_VOLTAGE_PATH, ADC_VOLTAGE_SUB_PATH, oCpt::protocol::userspace::fileExist(), and oCpt::protocol::userspace::modLoaded().

Here is the call graph for this function:



6.3.3.2 ~adc()

`oCpt::protocol::adc::~adc () [virtual]`

The deconstructor

Definition at line 79 of file Controller.cpp.

6.3.4 Member Function Documentation

6.3.4.1 compare()

```
bool oCpt::protocol::adc::compare (
    const uint8_t & id,
    const uint8_t & device = 0 )
```

Compare function

Parameters

<i>id</i>	ID to be checked
<i>device</i>	Device name to be checked

Returns

either true or false

Definition at line 95 of file Controller.cpp.

6.3.4.2 `getValue()`

```
uint16_t & oCpt::protocol::adc::getValue ( )
```

gets the current raw voltage level as resolution

Returns

the raw voltage level as uint16_t

Definition at line 81 of file Controller.cpp.

6.3.4.3 `operator==()`

```
bool oCpt::protocol::adc::operator== ( const adc & rhs )
```

Checks if adc object is the same

Parameters

<i>rhs</i>	other adc object, to be checked against
------------	---

Returns

either true or false

Definition at line 91 of file Controller.cpp.

References path_.

6.3.5 Member Data Documentation

6.3.5.1 `device_`

```
uint8_t oCpt::protocol::adc::device_ = 0 [private]
```

Definition at line 130 of file Controller.h.

6.3.5.2 `id_`

```
uint8_t oCpt::protocol::adc::id_ = 0 [private]
```

Definition at line 129 of file Controller.h.

6.3.5.3 path_

```
std::string oCpt::protocol::adc::path_ = "" [private]
```

Definition at line 131 of file Controller.h.

Referenced by operator==().

6.3.5.4 value_

```
uint16_t oCpt::protocol::adc::value_ = 0 [private]
```

Definition at line 132 of file Controller.h.

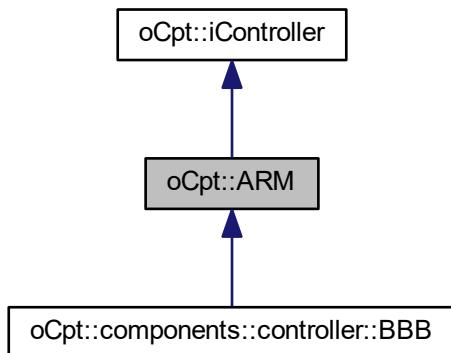
The documentation for this class was generated from the following files:

- /projects/mti/ohCaptain/ohCaptain/include/Core/[Controller.h](#)
- /projects/mti/ohCaptain/ohCaptain/src/Core/[Controller.cpp](#)

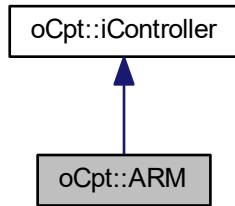
6.4 oCpt::ARM Class Reference

```
#include <Controller.h>
```

Inheritance diagram for oCpt::ARM:



Collaboration diagram for oCpt::ARM:



Public Member Functions

- [ARM \(World::ptr world\)](#)
- virtual [~ARM \(\)](#)
- virtual std::vector< protocol::adc::ptr > * [getAdcVector \(\)](#)
- virtual [protocol::adc::ptr getADC \(uint8_t id, uint8_t device\)](#)

Additional Inherited Members

6.4.1 Detailed Description

An [ARM](#) like controller. Currently only [ARM](#) devices are implemented

Definition at line 597 of file Controller.h.

6.4.2 Constructor & Destructor Documentation

6.4.2.1 ARM()

```
oCpt::ARM::ARM (
    World::ptr world )
```

The constructor of an [ARM](#) controller

Parameters

<code>world</code>	a shared_ptr to the World
--------------------	---

Definition at line 469 of file Controller.cpp.

6.4.2.2 ~ARM()

```
oCpt::ARM::~ARM ( ) [virtual]
```

The deconstructor

Definition at line 472 of file Controller.cpp.

6.4.3 Member Function Documentation

6.4.3.1 getADC()

```
protocol::adc::ptr oCpt::ARM::getADC (
    uint8_t id,
    uint8_t device ) [virtual]
```

Get a specific shared_ptr to an ADC

Parameters

<i>id</i>	the pin ID
<i>device</i>	the device ID

Returns

returns the specified ADC

Implements [oCpt::iController](#).

Definition at line 474 of file Controller.cpp.

References [oCpt::iController::adcVector_](#).

6.4.3.2 getAdcVector()

```
std::vector< protocol::adc::ptr > * oCpt::ARM::getAdcVector () [virtual]
```

Obtain a vector of available ADCs

Returns

Implements [oCpt::iController](#).

Definition at line 481 of file Controller.cpp.

References [oCpt::iController::adcVector_](#).

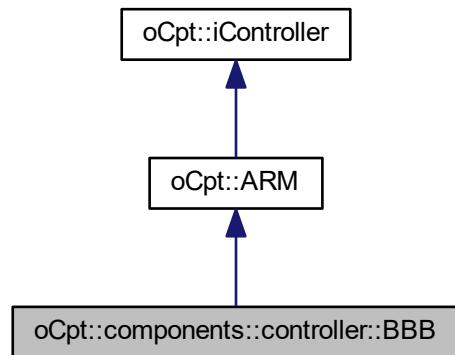
The documentation for this class was generated from the following files:

- /projects/mti/ohCaptain/ohCaptain/include/Core/[Controller.h](#)
- /projects/mti/ohCaptain/ohCaptain/src/Core/[Controller.cpp](#)

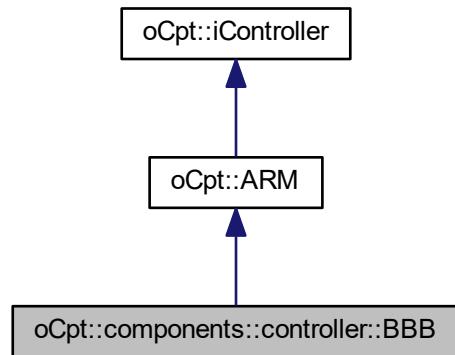
6.5 oCpt::components::controller::BBB Class Reference

```
#include <BeagleboneBlack.h>
```

Inheritance diagram for oCpt::components::controller::BBB:



Collaboration diagram for oCpt::components::controller::BBB:



Public Member Functions

- `BBB (World::ptr world)`
- virtual `~BBB ()`

Additional Inherited Members

6.5.1 Detailed Description

Definition at line 14 of file BeagleboneBlack.h.

6.5.2 Constructor & Destructor Documentation

6.5.2.1 BBB()

```
oCpt::components::controller::BBB::BBB ( 
    World::ptr world )
```

Definition at line 11 of file BeagleboneBlack.cpp.

References oCpt::iController::adcVector_.

6.5.2.2 ~BBB()

```
oCpt::components::controller::BBB::~BBB ( ) [virtual]
```

Definition at line 20 of file BeagleboneBlack.cpp.

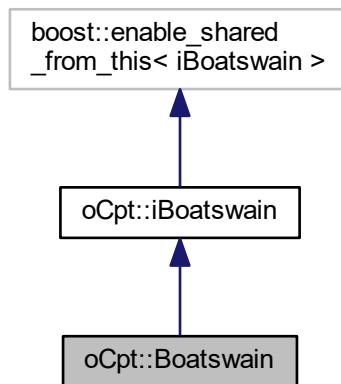
The documentation for this class was generated from the following files:

- /projects/mti/ohCaptain/ohCaptain/include/Controllers/[BeagleboneBlack.h](#)
- /projects/mti/ohCaptain/ohCaptain/src/Controllers/[BeagleboneBlack.cpp](#)

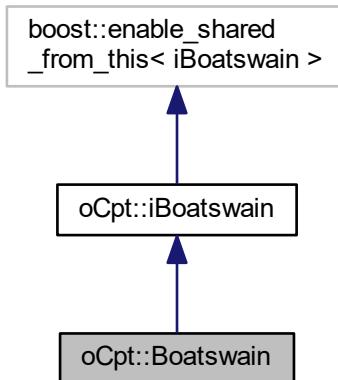
6.6 oCpt::Boatswain Class Reference

```
#include <Boatswain.h>
```

Inheritance diagram for oCpt::Boatswain:



Collaboration diagram for oCpt::Boatswain:



Public Member Functions

- `Boatswain (iController::ptr controller)`
- `virtual ~Boatswain () override`
- `virtual void run () override`
- `virtual void stop () override`
- `virtual void initialize () override`
- `virtual void registerSensor (iSensor::ptr sensor) override`
- `virtual void registerActuator (iActuator::ptr actuator) override`
- `virtual void registerComm (iComm::ptr comm) override`

Protected Member Functions

- `void resetTimer (iSensor::ptr sensor) override`

Additional Inherited Members

6.6.1 Detailed Description

The **Boatswain** performs all the labours tasks, suchs updateing and interpreting sensor readings, setting actuators according to the **Captain** wishes, updating the state representation of the vessel in the **World**. Each **Boatswain** runs on its own thread. It is possible for a vessel to have multiple Boatswains, responsible for multiple tasks, such as communication, localization, steering. Each **Boatswain** has to adhere to the **iBoatswain** interface.

Definition at line 113 of file `Boatswain.h`.

6.6.2 Constructor & Destructor Documentation

6.6.2.1 Boatswain()

```

oCpt::Boatswain::Boatswain (
    iController::ptr controller )
  
```

The constructor for a **Boatswain**

Parameters

<i>controller</i>	a shared_ptr to the controller with which teh Boatswain interacts
-------------------	---

Definition at line 29 of file Boatswain.cpp.

6.6.2.2 ~Boatswain()

```
oCpt::Boatswain::~Boatswain ( ) [override], [virtual]
```

Deconstructor

Definition at line 33 of file Boatswain.cpp.

6.6.3 Member Function Documentation

6.6.3.1 initialize()

```
void oCpt::Boatswain::initialize ( ) [override], [virtual]
```

Initialize the [Boatswain](#)

Implements [oCpt::iBoatswain](#).

Definition at line 48 of file Boatswain.cpp.

6.6.3.2 registerActuator()

```
void oCpt::Boatswain::registerActuator (
    iActuator::ptr actuator ) [override], [virtual]
```

Register a new [Actuator](#) with the [Boatswain](#)

Parameters

<i>actuator</i>	a shared_ptr to an Actuator
-----------------	---

Implements [oCpt::iBoatswain](#).

Definition at line 71 of file Boatswain.cpp.

6.6.3.3 registerComm()

```
void oCpt::Boatswain::registerComm (
    iComm::ptr comm ) [override], [virtual]
```

Register a new [iComm](#) device by setting a shared IO service

Parameters

<code>comm</code>	<input type="text"/>
-------------------	----------------------

Implements [oCpt::iBoatswain](#).

Definition at line 100 of file [Boatswain.cpp](#).

References [oCpt::iBoatswain::ioservice_](#).

6.6.3.4 registerSensor()

```
void oCpt::Boatswain::registerSensor (
    iSensor::ptr sensor ) [override], [virtual]
```

Register a new [Sensor](#) with the [Boatswain](#). If the Timer for the [Sensor](#) is set to a value greater than 0, the [Sensor](#) is registered with the [timerSensors_](#) and a timer is set. Otherwise the [Sensor](#) is registered as a [manualSensors_](#).

Parameters

<code>sensor</code>	a shared_ptr to a Sensor
---------------------	--

If the timer is set for the sensor, create a new timer service, register the sensor with the timer sensors, and set the callback functions to execute the [Sensor::run](#) function and the internal [resetTimer](#) function. If the timer is not set register the Sensor with the manual sensor.

Implements [oCpt::iBoatswain](#).

Definition at line 52 of file [Boatswain.cpp](#).

References [oCpt::iBoatswain::ioservice_](#), [oCpt::iBoatswain::manualSensors_](#), [resetTimer\(\)](#), [oCpt::iSensor::run\(\)](#), [oCpt::iBoatswain::timers_](#), and [oCpt::iBoatswain::timerSensors_](#).

Here is the call graph for this function:

**6.6.3.5 resetTimer()**

```
void oCpt::Boatswain::resetTimer (
    iSensor::ptr sensor ) [override], [protected], [virtual]
```

reset the timer of a [Sensor](#)

Parameters

<code>sensor</code>	a shared_ptr to the Sensor
---------------------	--

Don't execute if the thread is stopped

Find the current index of the sensor, this could maybe optimized by using a mapping list

Set the new timer. drift isn't taken into account at the current time.

Implements [oCpt::iBoatswain](#).

Definition at line 75 of file Boatswain.cpp.

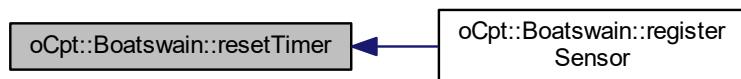
References `oCpt::iBoatswain::ioservice_`, `oCpt::iBoatswain::localStopThread_`, `oCpt::iSensor::run()`, `oCpt::iBoatswain::stopThread_`, `oCpt::iBoatswain::timers_`, and `oCpt::iBoatswain::timerSensors_`.

Referenced by `registerSensor()`.

Here is the call graph for this function:



Here is the caller graph for this function:



6.6.3.6 run()

```
void oCpt::Boatswain::run ( ) [override], [virtual]
```

Make the [Boatswain](#) work and execute the actuators, sensors and communications

Implements [oCpt::iBoatswain](#).

Definition at line 37 of file Boatswain.cpp.

References `oCpt::iBoatswain::ioservice_`, and `oCpt::iBoatswain::manualSensors_`.

6.6.3.7 stop()

```
void oCpt::Boatswain::stop ( ) [override], [virtual]
```

Stop the execution of the tasks

Implements [oCpt::iBoatswain](#).

Definition at line 44 of file Boatswain.cpp.

References [oCpt::iBoatswain::localStopThread_](#).

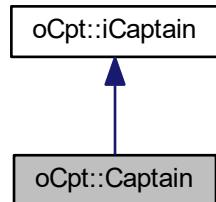
The documentation for this class was generated from the following files:

- /projects/mti/ohCaptain/ohCaptain/include/Core/[Boatswain.h](#)
- /projects/mti/ohCaptain/ohCaptain/src/Core/[Boatswain.cpp](#)

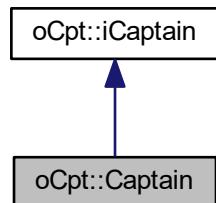
6.7 oCpt::Captain Class Reference

```
#include <Captain.h>
```

Inheritance diagram for oCpt::Captain:



Collaboration diagram for oCpt::Captain:



Public Member Functions

- [Captain \(World::ptr world\)](#)
- virtual [~Captain \(\) override](#)
- virtual void [run \(\) override](#)
- virtual void [stop \(\) override](#)
- virtual void [initialize \(\) override](#)

Additional Inherited Members

6.7.1 Detailed Description

Definition at line 36 of file Captain.h.

6.7.2 Constructor & Destructor Documentation

6.7.2.1 Captain()

```
oCpt::Captain::Captain (
    World::ptr world )
```

Definition at line 9 of file Captain.cpp.

References [oCpt::iCaptain::localStopThread_](#).

6.7.2.2 ~Captain()

```
oCpt::Captain::~Captain ( ) [override], [virtual]
```

Definition at line 13 of file Captain.cpp.

6.7.3 Member Function Documentation

6.7.3.1 initialize()

```
void oCpt::Captain::initialize ( ) [override], [virtual]
```

Implements [oCpt::iCaptain](#).

Definition at line 27 of file Captain.cpp.

6.7.3.2 run()

```
void oCpt::Captain::run ( ) [override], [virtual]
```

Implements [oCpt::iCaptain](#).

Definition at line 17 of file Captain.cpp.

References [oCpt::iCaptain::localStopThread_](#), and [oCpt::iCaptain::stopThread_](#).

6.7.3.3 stop()

```
void oCpt::Captain::stop ( ) [override], [virtual]
```

Implements [oCpt::iCaptain](#).

Definition at line 23 of file Captain.cpp.

References [oCpt::iCaptain::localStopThread_](#).

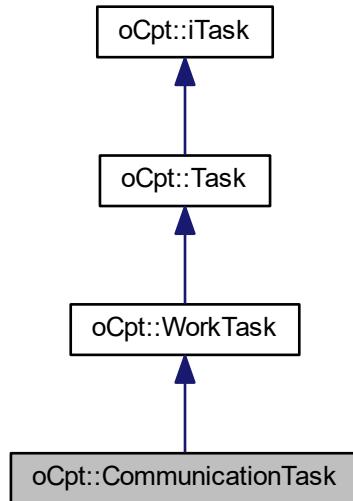
The documentation for this class was generated from the following files:

- /projects/mti/ohCaptain/ohCaptain/include/Core/[Captain.h](#)
- /projects/mti/ohCaptain/ohCaptain/src/Core/[Captain.cpp](#)

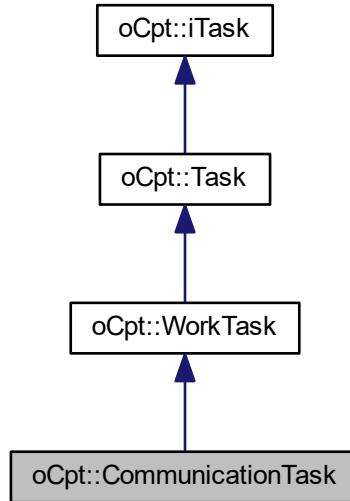
6.8 oCpt::CommunicationTask Class Reference

```
#include <Task.h>
```

Inheritance diagram for oCpt::CommunicationTask:



Collaboration diagram for oCpt::CommunicationTask:



Public Member Functions

- [CommunicationTask \(`Vessel::ptr vessel, bool concurrent=true`\)](#)
- virtual [~CommunicationTask \(\)](#)

Additional Inherited Members

6.8.1 Detailed Description

Definition at line 322 of file Task.h.

6.8.2 Constructor & Destructor Documentation

6.8.2.1 CommunicationTask()

```
oCpt::CommunicationTask::CommunicationTask (
    Vessel::ptr vessel,
    bool concurrent = true )
```

Definition at line 81 of file Task.cpp.

6.8.2.2 ~CommunicationTask()

```
oCpt::CommunicationTask::~CommunicationTask ( ) [virtual]
```

Definition at line 83 of file Task.cpp.

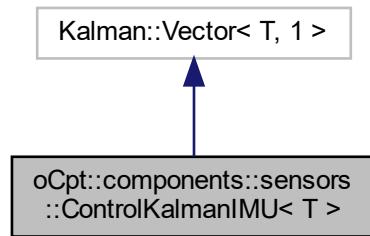
The documentation for this class was generated from the following files:

- /projects/mti/ohCaptain/ohCaptain/include/Core/Task.h
- /projects/mti/ohCaptain/ohCaptain/src/Core/Task.cpp

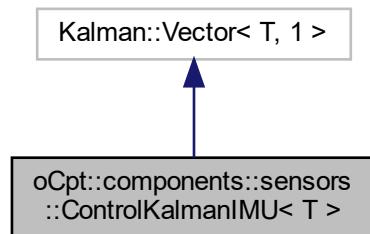
6.9 oCpt::components::sensors::ControlKalmanIMU< T > Class Template Reference

```
#include <KalmanIMU.h>
```

Inheritance diagram for oCpt::components::sensors::ControlKalmanIMU< T >:



Collaboration diagram for oCpt::components::sensors::ControlKalmanIMU< T >:



Public Types

- `typedef Kalman::Vector< T, 1 > Base`

Public Member Functions

- [ControlKalmanIMU \(void\)](#)
- [template<typename OtherDerived > ControlKalmanIMU \(const Eigen::MatrixBase< OtherDerived > &other\)](#)
- [template<typename OtherDerived > ControlKalmanIMU & operator= \(const Eigen::MatrixBase< OtherDerived > &other\)](#)

6.9.1 Detailed Description

```
template<typename T>
class oCpt::components::sensors::ControlKalmanIMU< T >
```

Definition at line 100 of file KalmanIMU.h.

6.9.2 Member Typedef Documentation

6.9.2.1 Base

```
template<typename T >
typedef Kalman::Vector< T , 1 > oCpt::components::sensors::ControlKalmanIMU< T >::Base
```

Definition at line 102 of file KalmanIMU.h.

6.9.3 Constructor & Destructor Documentation

6.9.3.1 ControlKalmanIMU() [1/2]

```
template<typename T >
oCpt::components::sensors::ControlKalmanIMU< T >::ControlKalmanIMU (
    void ) [inline]
```

Definition at line 102 of file KalmanIMU.h.

6.9.3.2 ControlKalmanIMU() [2/2]

```
template<typename T >
template<typename OtherDerived >
oCpt::components::sensors::ControlKalmanIMU< T >::ControlKalmanIMU (
    const Eigen::MatrixBase< OtherDerived > & other ) [inline]
```

Definition at line 102 of file KalmanIMU.h.

6.9.4 Member Function Documentation

6.9.4.1 operator=()

```
template<typename T >
template<typename OtherDerived >
ControlKalmanIMU& oCpt::components::sensors::ControlKalmanIMU< T >::operator= (
    const Eigen::MatrixBase< OtherDerived > & other ) [inline]
```

Definition at line 102 of file KalmanIMU.h.

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

- /projects/mti/ohCaptain/ohCaptain/include/Sensors/KalmanIMU.h

6.10 oCpt::World::Location::coordinate Struct Reference

```
#include <World.h>
```

Public Attributes

- `degree_t` `value`
- `cardinal_direction` `direction`

6.10.1 Detailed Description

Definition at line 134 of file World.h.

6.10.2 Member Data Documentation

6.10.2.1 direction

```
cardinal_direction oCpt::World::Location::coordinate::direction
```

Definition at line 136 of file World.h.

Referenced by `oCpt::components::sensors::Gps::interpretMsg()`.

6.10.2.2 value

```
degree_t oCpt::World::Location::coordinate::value
```

Definition at line 135 of file World.h.

Referenced by `oCpt::components::sensors::Gps::interpretMsg()`.

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

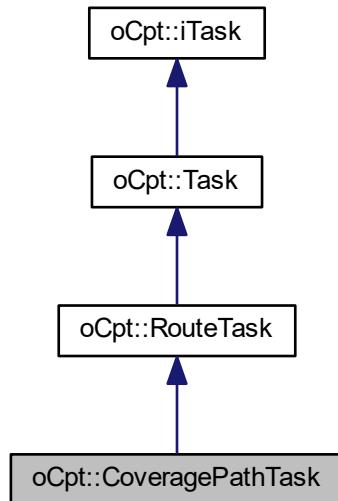
- /projects/mti/ohCaptain/ohCaptain/include/Core/World.h

6.11 oCpt::CoveragePathTask Class Reference

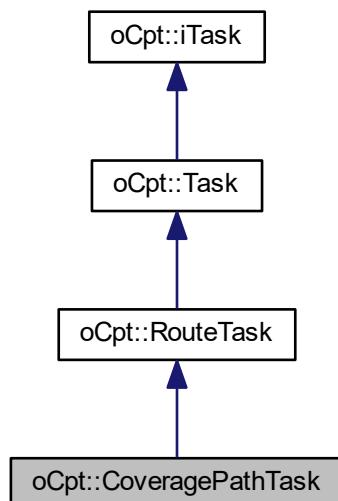
An object representing a coverage path task.

```
#include <Task.h>
```

Inheritance diagram for oCpt::CoveragePathTask:



Collaboration diagram for oCpt::CoveragePathTask:



Public Member Functions

- `CoveragePathTask (Vessel::ptr vessel, bool concurrent=false)`
- virtual `~CoveragePathTask ()`

Additional Inherited Members

6.11.1 Detailed Description

An object representing a coverage path task.

All these types of tasks need a robot to cover a complete region in order to perform their tasks. According to {cao_region_1988} such a mobile robot should use the following criteria, for a region filling operation:

1. The mobile robot must move through an entire area, i.e., the overall travel must cover a whole region.
2. The mobile robot must fill the region without overlapping paths.
3. Continuous and sequential operations without any repetition of paths is required of the robot.
4. The robot must avoid all obstacles in a region.
5. Simple motion trajectories (e.g., straight lines or circles) should be used for simplicity in control.
6. An "optimal" path is desired under the available conditions. It is not always possible to satisfy all these criteria for a complex environment. Sometimes a priority consideration is required.

Definition at line 201 of file Task.h.

6.11.2 Constructor & Destructor Documentation

6.11.2.1 CoveragePathTask()

```
oCpt::CoveragePathTask::CoveragePathTask (
    Vessel::ptr vessel,
    bool concurrent = false )
```

Constructor of the interface

Returns

Definition at line 53 of file Task.cpp.

6.11.2.2 ~CoveragePathTask()

```
oCpt::CoveragePathTask::~CoveragePathTask ( ) [virtual]
```

The deconstructor

Definition at line 55 of file Task.cpp.

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

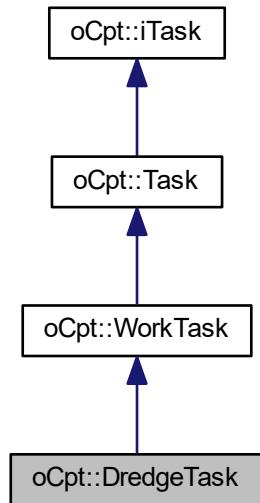
- /projects/mti/ohCaptain/ohCaptain/include/Core/[Task.h](#)
- /projects/mti/ohCaptain/ohCaptain/src/Core/[Task.cpp](#)

6.12 oCpt::DredgeTask Class Reference

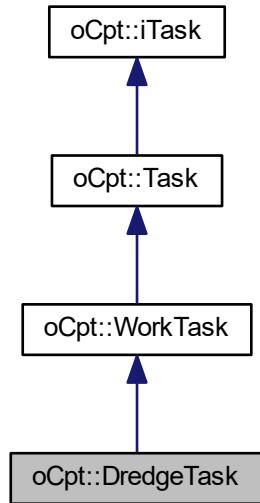
An Object representing a dredging task.

```
#include <Task.h>
```

Inheritance diagram for oCpt::DredgeTask:



Collaboration diagram for oCpt::DredgeTask:



Public Member Functions

- [DredgeTask \(Vessel::ptr vessel, bool concurrent=true\)](#)
- virtual [~DredgeTask \(\)](#)

Additional Inherited Members

6.12.1 Detailed Description

An Object representing a dredging task.

All these types tasks make use of an actuator and sensors to perform dredging tasks

Definition at line 287 of file Task.h.

6.12.2 Constructor & Destructor Documentation

6.12.2.1 DredgeTask()

```
oCpt::DredgeTask::DredgeTask (
    Vessel::ptr vessel,
    bool concurrent = true )
```

Constructor of the interface

Returns

Definition at line 69 of file Task.cpp.

6.12.2.2 ~DredgeTask()

```
oCpt::DredgeTask::~DredgeTask ( ) [virtual]
```

The deconstructor

Definition at line 71 of file Task.cpp.

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

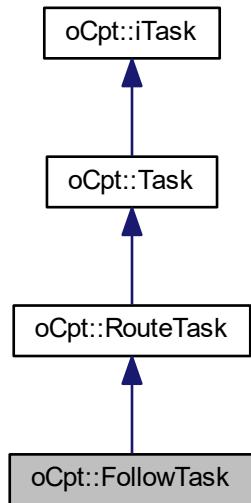
- [/projects/mti/ohCaptain/ohCaptain/include/Core/Task.h](#)
- [/projects/mti/ohCaptain/ohCaptain/src/Core/Task.cpp](#)

6.13 oCpt::FollowTask Class Reference

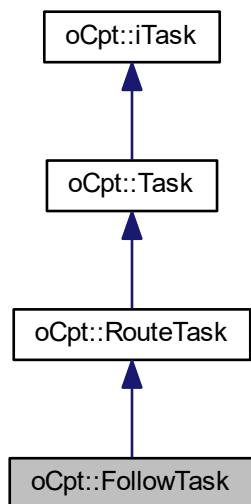
An object representing a follow the target task.

```
#include <Task.h>
```

Inheritance diagram for oCpt::FollowTask:



Collaboration diagram for oCpt::FollowTask:



Public Member Functions

- [FollowTask \(`Vessel::ptr` vessel, bool concurrent=false\)](#)
- virtual [~FollowTask \(\)](#)

Additional Inherited Members

6.13.1 Detailed Description

An object representing a follow the target task.

All these types of tasks need to follow a (moving) target

Definition at line 222 of file Task.h.

6.13.2 Constructor & Destructor Documentation

6.13.2.1 FollowTask()

```
oCpt::FollowTask::FollowTask (
    Vessel::ptr vessel,
    bool concurrent = false )
```

Constructor of the interface

Returns

Definition at line 57 of file Task.cpp.

6.13.2.2 ~FollowTask()

```
oCpt::FollowTask::~FollowTask ( ) [virtual]
```

The deconstructor

Definition at line 59 of file Task.cpp.

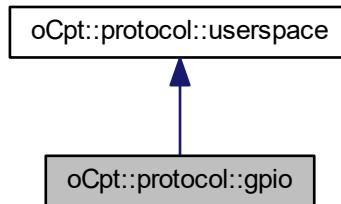
The documentation for this class was generated from the following files:

- [/projects/mti/ohCaptain/ohCaptain/include/Core/Task.h](#)
- [/projects/mti/ohCaptain/ohCaptain/src/Core/Task.cpp](#)

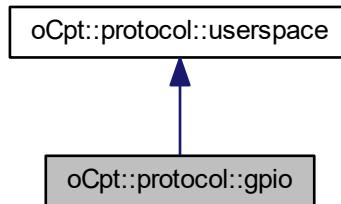
6.14 oCpt::protocol::gpio Class Reference

```
#include <Controller.h>
```

Inheritance diagram for oCpt::protocol::gpio:



Collaboration diagram for oCpt::protocol::gpio:



Public Types

- enum `Direction` { `INPUT` = 105, `OUTPUT` = 111 }
- enum `Value` { `LOW` = 48, `HIGH` = 49 }
- enum `Edge` { `NONE` = 110, `RISING` = 114, `FALLING` = 102, `BOTH` = 98 }
- typedef boost::shared_ptr<`gpio`> `ptr`
- typedef boost::signals2::signal<void()> `signal_t`
- typedef std::function<void()> `cb_func`

Public Member Functions

- `gpio` (int pinNumber, `Direction` direction=`INPUT`, `Value` value=`LOW`, `Edge` edge=`NONE`)
- `~gpio` ()
- int `getPinNumber` () const
- void `setPinNumber` (int pinNumber)

- `Value getValue () const`
- `void setValue (Value value)`
- `Direction getDirection () const`
- `void setDirection (Direction direction)`
- `Edge getEdge () const`
- `void setEdge (Edge edge)`
- `void setCallbackFunction (cb_func cb)`
- `void waitForEdge ()`
- `void waitForEdgeAsync ()`
- `void toggle ()`

Static Public Member Functions

- `static std::vector< ptr > exportedGpios ()`

Public Attributes

- `signal_t signalChanged`

Private Member Functions

- `void internalCbFunc ()`
- `void exportPin (const int &number)`
- `void unexportPin (const int &number)`
- `template<typename T >`
`void writePinValue (const int &number, const T &value)`
- `template<typename T >`
`void writePinValue (std::string path, const T &value)`

Static Private Member Functions

- `template<typename T >`
`static T readPinValue (const int &number)`
- `template<typename T >`
`static T readPinValue (std::string path)`

Private Attributes

- `int pinNumber_`
- `Value value_`
- `Direction direction_`
- `Edge edge_`
- `std::string gpiopath_`
- `cb_func cb_`
- `bool threadRunning_`

Additional Inherited Members

6.14.1 Detailed Description

A General Pin Input Output class. This is the class that handles gpio's in user space. Each pin can be set as either input or output, and have a High or a Low out-/input. When a pin is set as input, it can be polled on the edge, execute a function or send a signal on the rising, falling or changing edge of the signal

Definition at line 138 of file Controller.h.

6.14.2 Member Typedef Documentation

6.14.2.1 cb_func

```
typedef std::function<void()> oCpt::protocol::gpio::cb_func
```

Definition at line 142 of file Controller.h.

6.14.2.2 ptr

```
typedef boost::shared_ptr<gpio> oCpt::protocol::gpio::ptr
```

Definition at line 140 of file Controller.h.

6.14.2.3 signal_t

```
typedef boost::signals2::signal<void()> oCpt::protocol::gpio::signal_t
```

Definition at line 141 of file Controller.h.

6.14.3 Member Enumeration Documentation

6.14.3.1 Direction

```
enum oCpt::protocol::gpio::Direction
```

The Direction of the pin

Enumerator

INPUT	
OUTPUT	

Definition at line 147 of file Controller.h.

6.14.3.2 Edge

```
enum oCpt::protocol::gpio::Edge
```

Enumerator

NONE	
RISING	
FALLING	
BOTH	

Definition at line 157 of file Controller.h.

6.14.3.3 Value

```
enum oCpt::protocol::gpio::Value
```

Enumerator

LOW	
HIGH	

Definition at line 152 of file Controller.h.

6.14.4 Constructor & Destructor Documentation

6.14.4.1 gpio()

```
oCpt::protocol::gpio::gpio (
    int pinNumber,
    gpio::Direction direction = INPUT,
    gpio::Value value = LOW,
    gpio::Edge edge = NONE )
```

The constructor for the gpio class

Parameters

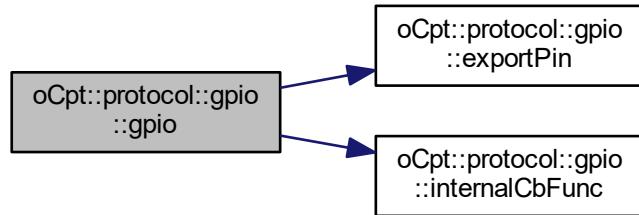
<i>pinNumber</i>	the pin pumber (in user-space mapping)
<i>direction</i>	the Direction of a pin, with a default value as Direction::INPUT
<i>value</i>	the start Value of a pin. With a default value of Value::LOW
<i>edge</i>	the Edge of a pin with the default value of Edge::NONE

Definition at line 337 of file Controller.cpp.

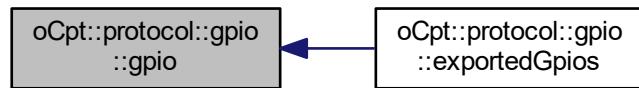
References cb_, direction_, edge_, exportPin(), GPIO_BASE_PATH, gpiopath_, internalCbFunc(), pinNumber_, and value_.

Referenced by `exportedGpios()`.

Here is the call graph for this function:



Here is the caller graph for this function:



6.14.4.2 ~gpio()

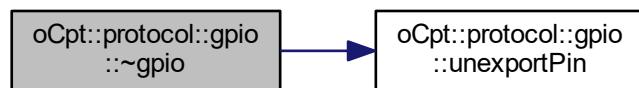
`oCpt::protocol::gpio::~gpio()`

The deconstructor

Definition at line 353 of file Controller.cpp.

References `pinNumber_`, and `unexportPin()`.

Here is the call graph for this function:



6.14.5 Member Function Documentation

6.14.5.1 exportedGpios()

```
std::vector< gpio::ptr > oCpt::protocol::gpio::exportedGpios ( ) [static]
```

Static function which creates a vector containing new gpio shared_ptr for each pin that is currently exported in the user space.

Returns

A vector with shared_ptr's of all exported gpio's in user-space

Iterate through all exported pins

Definition at line 357 of file Controller.cpp.

References gpio(), and GPIO_BASE_PATH.

Here is the call graph for this function:



6.14.5.2 exportPin()

```
void oCpt::protocol::gpio::exportPin (
    const int & number ) [private]
```

Export the pin in user-space

Parameters

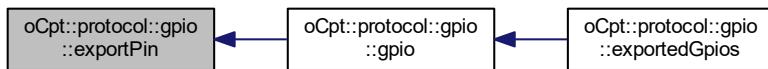
<i>number</i>	the pin number to be exported
---------------	-------------------------------

Definition at line 381 of file Controller.cpp.

References GPIO_BASE_PATH.

Referenced by gpio().

Here is the caller graph for this function:



6.14.5.3 getDirection()

```
gpio::Direction oCpt::protocol::gpio::getDirection() const
```

Get the current Direction. note this doesn't take into account external changes done outside this library

Returns

either Direction::INPUT or Direction::Output

Definition at line 314 of file Controller.cpp.

6.14.5.4 getEdge()

```
gpio::Edge oCpt::protocol::gpio::getEdge() const
```

Get the current Edge of the pin. If the Direction is set to Direction::INPUT the value is set in user-space, otherwise it is set in the object itself

Returns

Definition at line 323 of file Controller.cpp.

6.14.5.5 getPinNumber()

```
int oCpt::protocol::gpio::getPinNumber() const
```

Get the current pin number

Returns

an int representing the pin number in user-space mapping

Definition at line 291 of file Controller.cpp.

6.14.5.6 `getValue()`

```
gpio::Value oCpt::protocol::gpio::getValue ( ) const
```

Get the current value of the pin, if the Direction is set to Direction::INPUT the value is obtained from the user space, otherwise the value is read from object itself

Returns

either Value::HIGH or Value::LOW

Definition at line 300 of file Controller.cpp.

6.14.5.7 `internalCbFunc()`

```
void oCpt::protocol::gpio::internalCbFunc ( ) [private]
```

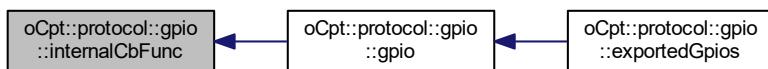
The internal callback function, which triggers the signalChanged signal

Definition at line 418 of file Controller.cpp.

References signalChanged.

Referenced by gpio().

Here is the caller graph for this function:



6.14.5.8 `readPinValue()` [1/2]

```
template<typename T >
static T oCpt::protocol::gpio::readPinValue (
    const int & number ) [inline], [static], [private]
```

Static generic function returning the value in user-space of either Direction, Edge or Value. Depending on the typename

Template Parameters

T	The value to return either the Value, Edge or Direction
-----	---

Parameters

<i>number</i>	the pin as number
---------------	-------------------

Returns

The read value as either Value, Edge or Direction

Definition at line 291 of file Controller.h.

References GPIO_BASE_PATH.

6.14.5.9 readPinValue() [2/2]

```
template<typename T >
static T oCpt::protocol::gpio::readPinValue (
    std::string path ) [inline], [static], [private]
```

Static generic function returning the value in user-space of either Direction, Edge or Value. Depending on the typename. This function is quicker then the overload function taking the pin number as int. and is therefore preferred to obtain the Value.

Template Parameters

<i>T</i>	The value to return either the Value, Edge or Direction
----------	---

Parameters

<i>path</i>	the pin as user-space path
-------------	----------------------------

Returns

the read value as either Value, Edge or Direction

Definition at line 304 of file Controller.h.

References boost::units::constants::c.

6.14.5.10 setCallbackFunction()

```
void oCpt::protocol::gpio::setCallbackFunction (
    gpio::cb_func cb )
```

Set a new Callbackfunction which is called on a certain Edge

Parameters

<i>cb</i>	the callback function
-----------	-----------------------

Definition at line 414 of file Controller.cpp.

References cb_.

6.14.5.11 setDirection()

```
void oCpt::protocol::gpio::setDirection (
    gpio::Direction direction )
```

Set the Direction of the pin

Parameters

<i>direction</i>	the Direction of the pin
------------------	--------------------------

Definition at line 318 of file Controller.cpp.

References direction_.

6.14.5.12 setEdge()

```
void oCpt::protocol::gpio::setEdge (
    gpio::Edge edge )
```

Set the Edge of the of the pin. if the Direction is set to Direction::INPUT, the value is set in user-space, otherwise it is set in the object itself

Parameters

<i>edge</i>	
-------------	--

Definition at line 330 of file Controller.cpp.

References edge_.

6.14.5.13 setPinNumber()

```
void oCpt::protocol::gpio::setPinNumber (
    int pinNumber )
```

Set the new pinnumber (don't use yet)

Parameters

<i>pinNumber</i>	the pinmuber to be set
------------------	------------------------

Definition at line 295 of file Controller.cpp.

References pinNumber_.

6.14.5.14 setValue()

```
void oCpt::protocol::gpio::setValue (
    gpio::Value value )
```

Set the current value, if the Direction is set to Direction::OUTPUT the value is set in userspace, either it is set to object itself

Parameters

value	<input type="text"/>
-------	----------------------

Definition at line 307 of file Controller.cpp.

References value_.

6.14.5.15 toggle()

```
void oCpt::protocol::gpio::toggle ( )
```

Toggle the value_ of the pin if Value::High then the value_ is set to Value::LOW

Definition at line 409 of file Controller.cpp.

References gpiopath_, and value_.

6.14.5.16 unexportPin()

```
void oCpt::protocol::gpio::unexportPin (
    const int & number ) [private]
```

Unexport the pin in user-space

Parameters

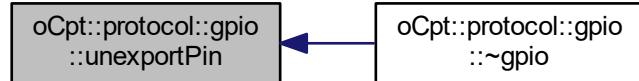
number	the pin number to be unexported
--------	---------------------------------

Definition at line 395 of file Controller.cpp.

References GPIO_BASE_PATH.

Referenced by ~gpio().

Here is the caller graph for this function:



6.14.5.17 waitForEdge()

```
void oCpt::protocol::gpio::waitForEdge( )
```

Wait for the occurrence of a change in Edge, corresponding with the set value of Edge. When the change is detected the callbackfunction is called. This function blocks the current thread.

Definition at line 422 of file Controller.cpp.

References cb_, direction_, and gpiopath_.

Referenced by waitForEdgeAsync().

Here is the caller graph for this function:



6.14.5.18 waitForEdgeAsync()

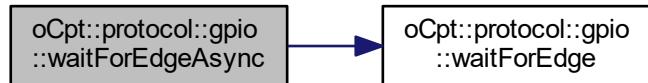
```
void oCpt::protocol::gpio::waitForEdgeAsync( )
```

Wait for the occurrence of a change in Edge, corresponding with the set value of Edge. When the chance is detected the callbackfunction is called. This function creates a new thread, allowing the current thread to run unhindered

Definition at line 457 of file Controller.cpp.

References waitForEdge().

Here is the call graph for this function:



6.14.5.19 writePinValue() [1/2]

```
template<typename T >
void oCpt::protocol::gpio::writePinValue (
    const int & number,
    const T & value ) [inline], [private]
```

Write the value to the pin. The T parameter determines which value to set

Template Parameters

<i>T</i>	the type could either be Value, Direction or Direction
----------	--

Parameters

<i>number</i>	the pin number as an integer
<i>value</i>	the Value to be set

Definition at line 331 of file Controller.h.

References GPIO_BASE_PATH.

6.14.5.20 writePinValue() [2/2]

```
template<typename T >
void oCpt::protocol::gpio::writePinValue (
    std::string path,
    const T & value ) [inline], [private]
```

Write the value to the pin, The T parameter determines which value to set. This overload is quicker than the one taking the integer and is therefore preferred

Template Parameters

<i>T</i>	the type could either be Value, Direction or Edge
----------	---

Parameters

<i>path</i>	the pin as an user-space path
<i>value</i>	the Value to write

Definition at line 344 of file Controller.h.

6.14.6 Member Data Documentation

6.14.6.1 cb_

`cb_func oCpt::protocol::gpio::cb_ [private]`

Definition at line 264 of file Controller.h.

Referenced by gpio(), setCallbackFunction(), and waitForEdge().

6.14.6.2 direction_

`Direction oCpt::protocol::gpio::direction_ [private]`

Definition at line 261 of file Controller.h.

Referenced by gpio(), setDirection(), and waitForEdge().

6.14.6.3 edge_

`Edge oCpt::protocol::gpio::edge_ [private]`

Definition at line 262 of file Controller.h.

Referenced by gpio(), and setEdge().

6.14.6.4 gpiopath_

`std::string oCpt::protocol::gpio::gpiopath_ [private]`

Definition at line 263 of file Controller.h.

Referenced by gpio(), toggle(), and waitForEdge().

6.14.6.5 pinNumber_

`int oCpt::protocol::gpio::pinNumber_ [private]`

Definition at line 259 of file Controller.h.

Referenced by gpio(), setPinNumber(), and ~gpio().

6.14.6.6 signalChanged

```
signal_t oCpt::protocol::gpio::signalChanged
```

The signal that is send if the internal callback function is executed

Definition at line 256 of file Controller.h.

Referenced by internalCbFunc().

6.14.6.7 threadRunning_

```
bool oCpt::protocol::gpio::threadRunning_ [private]
```

Definition at line 265 of file Controller.h.

6.14.6.8 value_

```
Value oCpt::protocol::gpio::value_ [private]
```

Definition at line 260 of file Controller.h.

Referenced by gpio(), setValue(), and toggle().

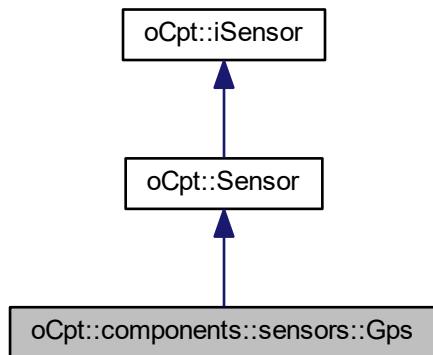
The documentation for this class was generated from the following files:

- /projects/mti/ohCaptain/ohCaptain/include/Core/[Controller.h](#)
- /projects/mti/ohCaptain/ohCaptain/src/Core/[Controller.cpp](#)

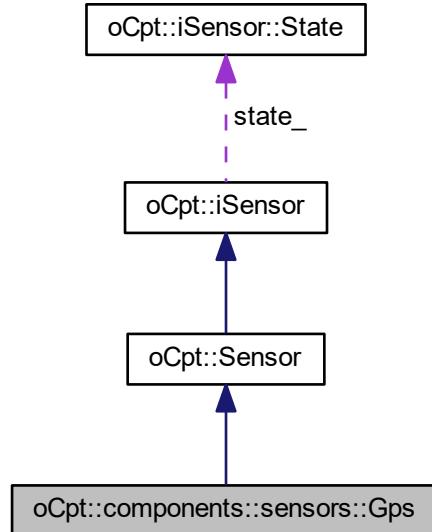
6.15 oCpt::components::sensors::Gps Class Reference

```
#include <Gps.h>
```

Inheritance diagram for oCpt::components::sensors::Gps:



Collaboration diagram for oCpt::components::sensors::Gps:



Public Types

- `typedef oCpt::World::Location::gpsPoint_t ReturnValue_t`

Public Member Functions

- `Gps (iController::ptr controller, World::ptr world, std::string id, std::string device, unsigned int baudrate)`
- `~Gps ()`
- `void updateSensor ()`
- `void run ()`
- `void stop ()`
- `void setIoservice (boost::shared_ptr< boost::asio::io_service > ioservice)`

Protected Member Functions

- `void interpretMsg ()`

Protected Attributes

- `std::string device_`
- `protocol::Serial::ptr serial_`

6.15.1 Detailed Description

Definition at line 13 of file Gps.h.

6.15.2 Member Typedef Documentation

6.15.2.1 ReturnValue_t

```
typedef oCpt::World::Location::gpsPoint_t oCpt::components::sensors::Gps::ReturnValue_t
```

Definition at line 15 of file Gps.h.

6.15.3 Constructor & Destructor Documentation

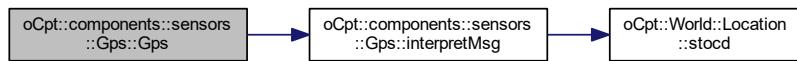
6.15.3.1 Gps()

```
oCpt::components::sensors::Gps::Gps (
    iController::ptr controller,
    World::ptr world,
    std::string id,
    std::string device,
    unsigned int baudrate )
```

Definition at line 13 of file Gps.cpp.

References interpretMsg(), and serial_.

Here is the call graph for this function:



6.15.3.2 ~Gps()

```
oCpt::components::sensors::Gps::~Gps ( )
```

Definition at line 28 of file Gps.cpp.

References serial_.

6.15.4 Member Function Documentation

6.15.4.1 interpretMsg()

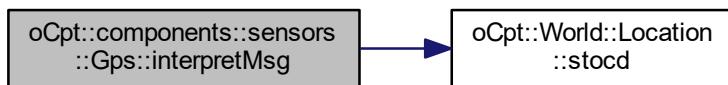
```
void oCpt::components::sensors::Gps::interpretMsg ( ) [protected]
```

Definition at line 59 of file Gps.cpp.

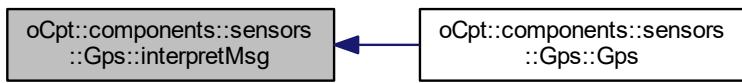
References oCpt::World::Location::coordinate::direction, oCpt::World::Location::gpsPoint::latitude, oCpt::World::Location::gpsPoint::longitude, serial_, oCpt::iSensor::sig_, oCpt::iSensor::State::Stamp, oCpt::iSensor::state_, oCpt::World::Location::stocd(), oCpt::iSensor::State::Value, oCpt::World::Location::coordinate::value, and oCpt::iSensor::world_.

Referenced by Gps().

Here is the call graph for this function:



Here is the caller graph for this function:



6.15.4.2 run()

```
void oCpt::components::sensors::Gps::run ( ) [virtual]
```

virtual function starting the run service for the IO

Reimplemented from [oCpt::Sensor](#).

Definition at line 38 of file Gps.cpp.

References oCpt::Sensor::run(), oCpt::iSensor::sensorRunning_, and serial_.

Here is the call graph for this function:



6.15.4.3 setIoservice()

```
void oCpt::components::sensors::Gps::setIoservice (
    boost::shared_ptr< boost::asio::io_service > ioservice ) [virtual]
```

Setting the used Asynchronous Input Output service

Parameters

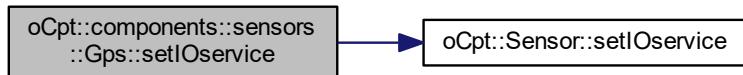
<i>ioservice</i>	ASIO IO service, which handles the async calls from multiple sensors
------------------	--

Reimplemented from [oCpt::Sensor](#).

Definition at line 54 of file Gps.cpp.

References serial_, and [oCpt::Sensor::setIoservice\(\)](#).

Here is the call graph for this function:



6.15.4.4 stop()

```
void oCpt::components::sensors::Gps::stop ( ) [virtual]
```

virtual function stopping the run

Reimplemented from [oCpt::Sensor](#).

Definition at line 45 of file Gps.cpp.

References `oCpt::iSensor::sensorRunning_`, `serial_`, and `oCpt::Sensor::stop()`.

Here is the call graph for this function:



6.15.4.5 updateSensor()

```
void oCpt::components::sensors::Gps::updateSensor ( ) [virtual]
```

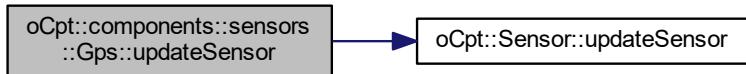
virtual function which performs a sensor update, obtaining a new value and sending a signal afterwards

Reimplemented from [oCpt::Sensor](#).

Definition at line 34 of file Gps.cpp.

References `oCpt::Sensor::updateSensor()`.

Here is the call graph for this function:



6.15.5 Member Data Documentation

6.15.5.1 device_

```
std::string oCpt::components::sensors::Gps::device_ [protected]
```

Definition at line 31 of file Gps.h.

6.15.5.2 serial_

```
protocol::Serial::ptr oCpt::components::sensors::Gps::serial_ [protected]
```

Definition at line 32 of file Gps.h.

Referenced by Gps(), interpretMsg(), run(), setIoservice(), stop(), and ~Gps().

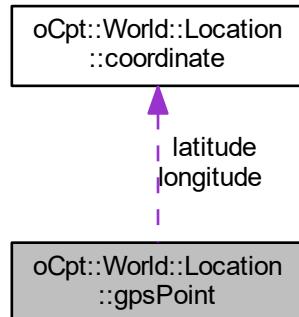
The documentation for this class was generated from the following files:

- /projects/mti/ohCaptain/ohCaptain/include/Sensors/[Gps.h](#)
- /projects/mti/ohCaptain/ohCaptain/src/Sensors/[Gps.cpp](#)

6.16 oCpt::World::Location::gpsPoint Struct Reference

```
#include <World.h>
```

Collaboration diagram for oCpt::World::Location::gpsPoint:



Public Member Functions

- std::string [toString \(\)](#)

Public Attributes

- `coordinate_t longitude`
- `coordinate_t latitude`
- `double height`

6.16.1 Detailed Description

Definition at line 139 of file World.h.

6.16.2 Member Function Documentation

6.16.2.1 `toString()`

```
std::string oCpt::World::Location::gpsPoint::toString ( )
```

Convert a gps coordinate to a text string

Returns

a text string eq. 5.000E,52.000N

Definition at line 72 of file World.cpp.

6.16.3 Member Data Documentation

6.16.3.1 `height`

```
double oCpt::World::Location::gpsPoint::height
```

Definition at line 142 of file World.h.

6.16.3.2 `latitude`

```
coordinate_t oCpt::World::Location::gpsPoint::latitude
```

Definition at line 141 of file World.h.

Referenced by oCpt::components::sensors::Gps::interpretMsg().

6.16.3.3 `longitude`

```
coordinate_t oCpt::World::Location::gpsPoint::longitude
```

Definition at line 140 of file World.h.

Referenced by oCpt::components::sensors::Gps::interpretMsg().

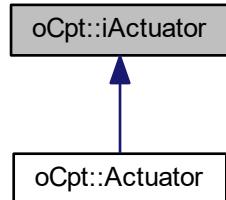
The documentation for this struct was generated from the following files:

- /projects/mti/ohCaptain/ohCaptain/include/Core/[World.h](#)
- /projects/mti/ohCaptain/ohCaptain/src/Core/[World.cpp](#)

6.17 oCpt::iActuator Class Reference

```
#include <Actuator.h>
```

Inheritance diagram for oCpt::iActuator:



Public Types

- `typedef boost::shared_ptr<iActuator> ptr`

Public Member Functions

- `iActuator()`
- `virtual ~iActuator()`
- `virtual void setActuator()=0`
- `virtual void run()=0`
- `virtual void stop()=0`

6.17.1 Detailed Description

Definition at line 17 of file Actuator.h.

6.17.2 Member Typedef Documentation

6.17.2.1 `ptr`

```
typedef boost::shared_ptr<iActuator> oCpt::iActuator::ptr
```

Definition at line 19 of file Actuator.h.

6.17.3 Constructor & Destructor Documentation

6.17.3.1 iActuator()

```
oCpt::iActuator::iActuator ( )
```

Definition at line 9 of file Actuator.cpp.

6.17.3.2 ~iActuator()

```
oCpt::iActuator::~iActuator ( ) [virtual]
```

Definition at line 13 of file Actuator.cpp.

6.17.4 Member Function Documentation

6.17.4.1 run()

```
virtual void oCpt::iActuator::run ( ) [pure virtual]
```

Implemented in [oCpt::Actuator](#).

6.17.4.2 setActuator()

```
virtual void oCpt::iActuator::setActuator ( ) [pure virtual]
```

Implemented in [oCpt::Actuator](#).

6.17.4.3 stop()

```
virtual void oCpt::iActuator::stop ( ) [pure virtual]
```

Implemented in [oCpt::Actuator](#).

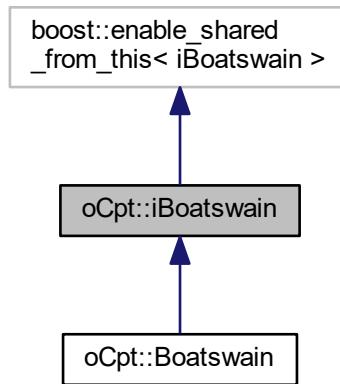
The documentation for this class was generated from the following files:

- /projects/mti/ohCaptain/ohCaptain/include/Core/[Actuator.h](#)
- /projects/mti/ohCaptain/ohCaptain/src/Core/[Actuator.cpp](#)

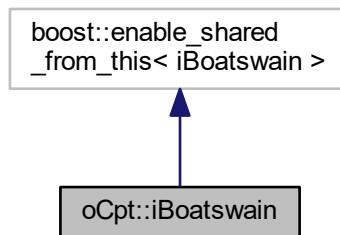
6.18 oCpt::iBoatswain Class Reference

```
#include <Boatswain.h>
```

Inheritance diagram for oCpt::iBoatswain:



Collaboration diagram for oCpt::iBoatswain:



Public Types

- `typedef boost::shared_ptr<iBoatswain> ptr`
- `typedef boost::shared_ptr<boost::asio::deadline_timer> timerPtr`

Public Member Functions

- `iBoatswain (iController::ptr controller)`
- `virtual ~iBoatswain ()`
- `virtual void run ()=0`
- `virtual void stop ()=0`
- `virtual void initialize ()=0`
- `virtual void registerSensor (iSensor::ptr sensor)=0`
- `virtual void registerActuator (iActuator::ptr actuator)=0`
- `virtual void registerComm (iComm::ptr comm)=0`
- `const boost::shared_ptr< bool > & getStopThread () const`
- `void setStopThread (const boost::shared_ptr< bool > &stopThread)`
- `boost::shared_ptr< boost::asio::io_service > & getIoservice ()`

Protected Member Functions

- `virtual void resetTimer (iSensor::ptr sensor)=0`

Protected Attributes

- `boost::shared_ptr< boost::asio::io_service > ioservice_`
- `iController::ptr controller_`
- `std::vector< timerPtr > timers_`
- `std::vector< iSensor::ptr > timerSensors_`
- `std::vector< iSensor::ptr > manualSensors_`
- `boost::shared_ptr< bool > stopThread_`
- `boost::shared_ptr< bool > localStopThread_`

6.18.1 Detailed Description

The `Boatswain` performs all the labours tasks, suchs updateing and interpretting sensor readings, setting actuators according to the `Captain` wishes, updating the state representation of the vessel in the `World`. Each `Boatswain` runs on its own thread. It is possible for a vessel to have multiple `Boatswains`, responsible for multiple tasks, such as communication, localization, steering. Each `Boatswain` has to adhere to the `iBoatswain` interface.

Definition at line 27 of file `Boatswain.h`.

6.18.2 Member Typedef Documentation

6.18.2.1 ptr

```
typedef boost::shared_ptr<iBoatswain> oCpt::iBoatswain::ptr
```

Definition at line 29 of file `Boatswain.h`.

6.18.2.2 timerPtr

```
typedef boost::shared_ptr<boost::asio::deadline_timer> oCpt::iBoatswain::timerPtr
```

Definition at line 30 of file `Boatswain.h`.

6.18.3 Constructor & Destructor Documentation

6.18.3.1 iBoatswain()

```
oCpt::iBoatswain::iBoatswain (
    iController::ptr controller )
```

Constructor for a [iBoatswain](#)

Parameters

<code>controller</code>	a shared_ptr to the controller with which teh Boatswain interacts
-------------------------	---

Definition at line 7 of file Boatswain.cpp.

References `ioservice_`, and `localStopThread_`.

6.18.3.2 ~iBoatswain()

```
oCpt::iBoatswain::~iBoatswain ( ) [virtual]
```

Deconstructor for the [iBoatswain](#)

Definition at line 13 of file Boatswain.cpp.

6.18.4 Member Function Documentation**6.18.4.1 getIOservice()**

```
boost::shared_ptr< boost::asio::io_service > & oCpt::iBoatswain::getIOservice ( )
```

get the used Input Output service

Returns

a shared_ptr to the ASIO io service

Definition at line 25 of file Boatswain.cpp.

References `ioservice_`.

6.18.4.2 getStopThread()

```
const boost::shared_ptr< bool > & oCpt::iBoatswain::getStopThread ( ) const
```

get if the thread is stopped

Returns

returns if the thread should stop

Definition at line 17 of file Boatswain.cpp.

References `stopThread_`.

6.18.4.3 initialize()

```
virtual void oCpt::iBoatswain::initialize ( ) [pure virtual]
```

pure virtual function of initializing the [Boatswain](#)

Implemented in [oCpt::Boatswain](#).

6.18.4.4 registerActuator()

```
virtual void oCpt::iBoatswain::registerActuator (
    iActuator::ptr actuator ) [pure virtual]
```

Pure virtual function for registering a new actuator with the [Boatswain](#)

Parameters

<i>actuator</i>	a shared_ptr to an Actuator which need to be maintained by the Boatswain
-----------------	--

Implemented in [oCpt::Boatswain](#).

6.18.4.5 registerComm()

```
virtual void oCpt::iBoatswain::registerComm (
    iComm::ptr comm ) [pure virtual]
```

Pure virtual function for registering a new communication device which

Parameters

<i>comm</i>	
-------------	--

Implemented in [oCpt::Boatswain](#).

6.18.4.6 registerSensor()

```
virtual void oCpt::iBoatswain::registerSensor (
    iSensor::ptr sensor ) [pure virtual]
```

Pure virtual function for registering a new sensor with the [Boatswain](#)

Parameters

<i>sensor</i>	a shared_ptr to a Sensor which need to be maintained by the Boatswain
---------------	---

Implemented in [oCpt::Boatswain](#).

6.18.4.7 resetTimer()

```
virtual void oCpt::iBoatswain::resetTimer (
    iSensor::ptr sensor ) [protected], [pure virtual]
```

Pure virtual function for resetting the timer

Parameters

<i>sensor</i>	
---------------	--

Implemented in [oCpt::Boatswain](#).

6.18.4.8 run()

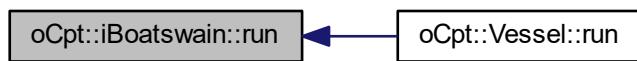
```
virtual void oCpt::iBoatswain::run ( ) [pure virtual]
```

pure virtual function for running the boatswain and his registered sensors

Implemented in [oCpt::Boatswain](#).

Referenced by [oCpt::Vessel::run\(\)](#).

Here is the caller graph for this function:



6.18.4.9 setStopThread()

```
void oCpt::iBoatswain::setStopThread ( const boost::shared_ptr< bool > & stopThread )
```

set the value of the stopthread

Parameters

<code>stopThread</code>	//<! a shared_ptr to the boolean
-------------------------	----------------------------------

Definition at line 21 of file Boatswain.cpp.

References `stopThread_`.

6.18.4.10 stop()

```
virtual void oCpt::iBoatswain::stop ( ) [pure virtual]
```

pure virtual function for stopping the run task

Implemented in [oCpt::Boatswain](#).

6.18.5 Member Data Documentation

6.18.5.1 controller_

```
iController::ptr oCpt::iBoatswain::controller_ [protected]
```

Definition at line 96 of file Boatswain.h.

6.18.5.2 ioservice_

```
boost::shared_ptr<boost::asio::io_service> oCpt::iBoatswain::ioservice_ [protected]
```

Definition at line 95 of file Boatswain.h.

Referenced by getIoservice(), iBoatswain(), oCpt::Boatswain::registerComm(), oCpt::Boatswain::registerSensor(), oCpt::Boatswain::resetTimer(), and oCpt::Boatswain::run().

6.18.5.3 localStopThread_

```
boost::shared_ptr<bool> oCpt::iBoatswain::localStopThread_ [protected]
```

Definition at line 101 of file Boatswain.h.

Referenced by iBoatswain(), oCpt::Boatswain::resetTimer(), and oCpt::Boatswain::stop().

6.18.5.4 manualSensors_

```
std::vector<iSensor::ptr> oCpt::iBoatswain::manualSensors_ [protected]
```

Definition at line 99 of file Boatswain.h.

Referenced by oCpt::Boatswain::registerSensor(), and oCpt::Boatswain::run().

6.18.5.5 stopThread_

```
boost::shared_ptr<bool> oCpt::iBoatswain::stopThread_ [protected]
```

Definition at line 100 of file Boatswain.h.

Referenced by getStopThread(), oCpt::Boatswain::resetTimer(), and setStopThread().

6.18.5.6 timers_

```
std::vector<timerPtr> oCpt::iBoatswain::timers_ [protected]
```

Definition at line 97 of file Boatswain.h.

Referenced by oCpt::Boatswain::registerSensor(), and oCpt::Boatswain::resetTimer().

6.18.5.7 timerSensors_

```
std::vector<iSensor::ptr> oCpt::iBoatswain::timerSensors_ [protected]
```

Definition at line 98 of file Boatswain.h.

Referenced by oCpt::Boatswain::registerSensor(), and oCpt::Boatswain::resetTimer().

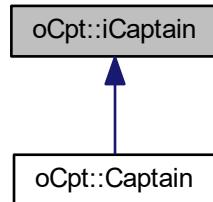
The documentation for this class was generated from the following files:

- /projects/mti/ohCaptain/ohCaptain/include/Core/[Boatswain.h](#)
- /projects/mti/ohCaptain/ohCaptain/src/Core/[Boatswain.cpp](#)

6.19 oCpt::iCaptain Class Reference

```
#include <Captain.h>
```

Inheritance diagram for oCpt::iCaptain:



Public Types

- `typedef boost::shared_ptr< iCaptain > ptr`

Public Member Functions

- `iCaptain (World::ptr world)`
- `virtual ~iCaptain ()`
- `virtual void run ()=0`
- `virtual void stop ()=0`
- `virtual void initialize ()=0`
- `const boost::shared_ptr< bool > & getStopThread_ () const`
- `void setStopThread_ (const boost::shared_ptr< bool > &stopThread_)`

Protected Attributes

- `boost::shared_ptr< bool > stopThread_`
- `boost::shared_ptr< bool > localStopThread_`
- `World::ptr world_`

6.19.1 Detailed Description

Definition at line 12 of file Captain.h.

6.19.2 Member Typedef Documentation

6.19.2.1 ptr

```
typedef boost::shared_ptr<iCaptain> oCpt::iCaptain::ptr
```

Definition at line 14 of file Captain.h.

6.19.3 Constructor & Destructor Documentation

6.19.3.1 iCaptain()

```
oCpt::iCaptain::iCaptain (
    World::ptr world )
```

Definition at line 31 of file Captain.cpp.

6.19.3.2 ~iCaptain()

```
oCpt::iCaptain::~iCaptain ( ) [virtual]
```

Definition at line 35 of file Captain.cpp.

6.19.4 Member Function Documentation

6.19.4.1 getStopThread_()

```
const boost::shared_ptr< bool > & oCpt::iCaptain::getStopThread_ ( ) const
```

Definition at line 39 of file Captain.cpp.

References stopThread_.

6.19.4.2 initialize()

```
virtual void oCpt::iCaptain::initialize ( ) [pure virtual]
```

Implemented in [oCpt::Captain](#).

6.19.4.3 run()

```
virtual void oCpt::iCaptain::run ( ) [pure virtual]
```

Implemented in [oCpt::Captain](#).

6.19.4.4 setStopThread_()

```
void oCpt::iCaptain::setStopThread_ (
    const boost::shared_ptr< bool > & stopThread_ )
```

Definition at line 43 of file Captain.cpp.

References stopThread_.

6.19.4.5 `stop()`

`virtual void oCpt::iCaptain::stop () [pure virtual]`

Implemented in [oCpt::Captain](#).

6.19.5 Member Data Documentation

6.19.5.1 `localStopThread_`

`boost::shared_ptr<bool> oCpt::iCaptain::localStopThread_ [protected]`

Definition at line 32 of file Captain.h.

Referenced by [oCpt::Captain::Captain\(\)](#), [oCpt::Captain::run\(\)](#), and [oCpt::Captain::stop\(\)](#).

6.19.5.2 `stopThread_`

`boost::shared_ptr<bool> oCpt::iCaptain::stopThread_ [protected]`

Definition at line 31 of file Captain.h.

Referenced by [getStopThread_\(\)](#), [oCpt::Captain::run\(\)](#), and [setStopThread_\(\)](#).

6.19.5.3 `world_`

`World::ptr oCpt::iCaptain::world_ [protected]`

Definition at line 33 of file Captain.h.

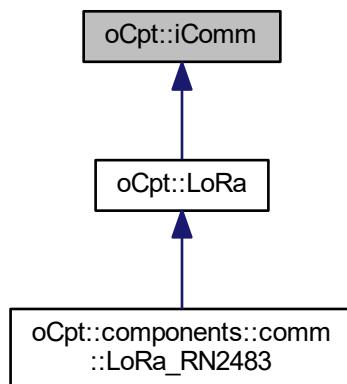
The documentation for this class was generated from the following files:

- /projects/mti/ohCaptain/ohCaptain/include/Core/[Captain.h](#)
- /projects/mti/ohCaptain/ohCaptain/src/Core/[Captain.cpp](#)

6.20 oCpt::iComm Class Reference

#include <Communication.h>

Inheritance diagram for oCpt::iComm:



Classes

- struct [Message](#)

Public Types

- typedef boost::shared_ptr< [iComm](#) > [ptr](#)
- typedef boost::signals2::signal< void()> [signal_t](#)

Public Member Functions

- [iComm](#) (const std::string &id, const std::string &device, [World::ptr](#) world=[World::ptr](#)(new [World\(\)](#)), [iController::io_t](#) ioservice=[iController::io_t](#)(new boost::asio::io_service()))
- virtual ~[iComm](#) ()
- virtual void [run](#) ()=0
- virtual void [stop](#) ()=0
- virtual void [initialize](#) ()=0
- virtual void [sendMessage](#) ([Message](#) msg)=0
- virtual [Message::ptr](#) [recieveMessage](#) ()=0
- virtual void [recieveAsyncMessage](#) ()=0
- const std::string & [getId](#) () const
- void [setId](#) (const std::string &id)
- const std::string & [getTypeOfComm](#) () const
- void [setTypeOfComm](#) (const std::string &typeOfComm)
- [Message::ptr](#) [readFiFoMsg](#) ()
- std::deque< [Message::ptr](#) > * [getMsgQueue](#) ()
- void [setioservice](#) (const [iController::io_t](#) &ioservice)

Public Attributes

- [signal_t](#) [msgRecievedSig](#)

Protected Attributes

- std::string [id_](#)
- std::string [typeOfComm_](#)
- std::string [device_](#)
- boost::posix_time::milliseconds [timer_](#)
- std::deque< [Message::ptr](#) > [msgQueue_](#)
- [iController::io_t](#) [ioservice_](#)
- [World::ptr](#) [world_](#)

6.20.1 Detailed Description

The interface for communication devices

Definition at line 26 of file Communication.h.

6.20.2 Member Typedef Documentation

6.20.2.1 ptr

```
typedef boost::shared_ptr<iComm> oCpt::iComm::ptr
```

Definition at line 28 of file Communication.h.

6.20.2.2 signal_t

```
typedef boost::signals2::signal<void()> oCpt::iComm::signal_t
```

Definition at line 29 of file Communication.h.

6.20.3 Constructor & Destructor Documentation

6.20.3.1 iComm()

```
oCpt::iComm::iComm (
    const std::string & id,
    const std::string & device,
    World::ptr world = World::ptr(new World()),
    iController::io_t ioservice = iController::io_t(new boost::asio::io_service()) )
```

The constructor for the communication interface

Parameters

<i>id</i>	The ID of the communication device
<i>device</i>	The device path eq. /dev/ttyS0
<i>world</i>	A shared_ptr to the world with a default to a newly created one
<i>ioservice</i>	A shared_ptr to an ASIO Input Output service with a newly created one as default

Definition at line 12 of file Communication.cpp.

6.20.3.2 ~iComm()

```
oCpt::iComm::~iComm ( ) [virtual]
```

The deconstructor

Definition at line 21 of file Communication.cpp.

6.20.4 Member Function Documentation

6.20.4.1 getId()

```
const std::string & oCpt::iComm::getId ( ) const
```

Returns the ID of the communication device

Returns

a string with the ID

Definition at line 25 of file Communication.cpp.

References id_.

6.20.4.2 getMsgQueue()

```
std::deque< iComm::Message::ptr > * oCpt::iComm::getMsgQueue ( )
```

A que with received [Message::ptr](#)

Returns

a pointer to the [Message](#) que

Definition at line 54 of file Communication.cpp.

References msgQueue_.

6.20.4.3 getTypeOfComm()

```
const std::string & oCpt::iComm::getTypeOfComm ( ) const
```

Get the type of communication device

Returns

a string with type of device eq. modem, serial, [LoRa](#), WiFi

Definition at line 33 of file Communication.cpp.

References typeOfComm_.

6.20.4.4 initialize()

```
virtual void oCpt::iComm::initialize ( ) [pure virtual]
```

A pure virtual function which initializes the communication device

Implemented in [oCpt::LoRa](#).

6.20.4.5 `readFiFoMsg()`

```
iComm::Message::ptr oCpt::iComm::readFiFoMsg ( )
```

Get a pointer to the first message in Queue

Returns

Definition at line 41 of file Communication.cpp.

References msgQueue_.

6.20.4.6 `recieveAsyncMessage()`

```
virtual void oCpt::iComm::recieveAsyncMessage ( ) [pure virtual]
```

A pure virtual function which performs the polling for a new message on a separate threads, so it won't block the current one, it needs to send a signal when the message is received

Implemented in [oCpt::LoRa](#).

6.20.4.7 `recieveMessage()`

```
virtual Message::ptr oCpt::iComm::recieveMessage ( ) [pure virtual]
```

A pure virtual function with a shared_ptr to the first in queue received message, this function will hold the current thread

Returns

a shared_ptr pointing towards the queued [Message](#)

Implemented in [oCpt::LoRa](#).

6.20.4.8 `run()`

```
virtual void oCpt::iComm::run ( ) [pure virtual]
```

a pure virtual function which runs the communication device

Implemented in [oCpt::LoRa](#).

6.20.4.9 `sendMessage()`

```
virtual void oCpt::iComm::sendMessage (
    Message msg ) [pure virtual]
```

A pure virtual function which sends the message

Parameters

<i>msg</i>	the Message , consisting of a payload and a time stamp
------------	--

Implemented in [oCpt::LoRa](#).

6.20.4.10 setId()

```
void oCpt::iComm::setId (
    const std::string & id )
```

Set the ID of the communication device

Parameters

<i>id</i>	The ID of the communication device
-----------	------------------------------------

Definition at line 29 of file Communication.cpp.

References [id_](#).

6.20.4.11 setIoservice()

```
void oCpt::iComm::setIoservice (
    const iController::io_t & ioservice )
```

The ASIO Input Output service handling the messages

Parameters

<i>ioservice</i>	a shared_ptr to a IO service
------------------	------------------------------

Definition at line 50 of file Communication.cpp.

References [ioservice_](#).

6.20.4.12 setTypeOfComm()

```
void oCpt::iComm::setTypeOfComm (
    const std::string & typeOfComm )
```

Set the type of communication device. eq. modem, serial, [LoRa](#), WiFi

Parameters

<i>typeOfComm</i>	string representing the type of communication
-------------------	---

Definition at line 37 of file Communication.cpp.

References typeOfComm_.

6.20.4.13 stop()

```
virtual void oCpt::iComm::stop () [pure virtual]
```

A pure virtual function which stops the communication device

Implemented in [oCpt::LoRa](#).

6.20.5 Member Data Documentation

6.20.5.1 device_

```
std::string oCpt::iComm::device_ [protected]
```

Definition at line 150 of file Communication.h.

6.20.5.2 id_

```
std::string oCpt::iComm::id_ [protected]
```

Definition at line 148 of file Communication.h.

Referenced by getId(), and setId().

6.20.5.3 ioservice_

```
iController::io_t oCpt::iComm::ioservice_ [protected]
```

Definition at line 153 of file Communication.h.

Referenced by setioservice().

6.20.5.4 msgQueue_

```
std::deque<Message::ptr> oCpt::iComm::msgQueue_ [protected]
```

Definition at line 152 of file Communication.h.

Referenced by getMsgQueue(), oCpt::LoRa::messageRecieved(), and readFiFoMsg().

6.20.5.5 msgRecievedSig

`signal_t oCpt::iComm::msgRecievedSig`

A signal which is send when a new [Message](#)

Definition at line 146 of file [Communication.h](#).

Referenced by [oCpt::LoRa::messageRecieved\(\)](#).

6.20.5.6 timer_

`boost::posix_time::milliseconds oCpt::iComm::timer_ [protected]`

Definition at line 151 of file [Communication.h](#).

6.20.5.7 typeOfComm_

`std::string oCpt::iComm::typeOfComm_ [protected]`

Definition at line 149 of file [Communication.h](#).

Referenced by [getTypeOfComm\(\)](#), and [setTypeOfComm\(\)](#).

6.20.5.8 world_

`World::ptr oCpt::iComm::world_ [protected]`

Definition at line 154 of file [Communication.h](#).

Referenced by [oCpt::LoRa::messageRecieved\(\)](#), and [oCpt::LoRa::sendMessage\(\)](#).

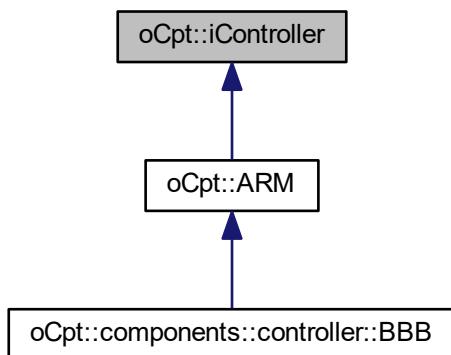
The documentation for this class was generated from the following files:

- /projects/mti/ohCaptain/ohCaptain/include/Core/[Communication.h](#)
- /projects/mti/ohCaptain/ohCaptain/src/Core/[Communication.cpp](#)

6.21 oCpt::iController Class Reference

#include <Controller.h>

Inheritance diagram for oCpt::iController:



Public Types

- `typedef boost::shared_ptr< iController > ptr`
- `typedef boost::shared_ptr< boost::asio::io_service > io_t`

Public Member Functions

- `iController (World::ptr world)`
- `virtual ~iController ()`
- `virtual std::vector< protocol::adc::ptr > * getAdcVector ()=0`
- `virtual protocol::adc::ptr getADC (uint8_t id, uint8_t device)=0`

Protected Attributes

- `std::vector< protocol::adc::ptr > adcVector_`
- `World::ptr world_`

6.21.1 Detailed Description

The interface for a controller. Each controller like for instance a Beaglebone black, Raspberry PI or x64 computer, should adhere to this interface

Definition at line 559 of file Controller.h.

6.21.2 Member Typedef Documentation

6.21.2.1 io_t

```
typedef boost::shared_ptr<boost::asio::io_service> oCpt::iController::io_t
```

Definition at line 562 of file Controller.h.

6.21.2.2 ptr

```
typedef boost::shared_ptr<iController> oCpt::iController::ptr
```

Definition at line 561 of file Controller.h.

6.21.3 Constructor & Destructor Documentation

6.21.3.1 iController()

```
oCpt::iController::iController (
    World::ptr world )
```

The constructor

Parameters

<i>world</i>	a pointer to the World
--------------	--

Definition at line 464 of file Controller.cpp.

6.21.3.2 ~iController()

```
oCpt::iController::~iController ( ) [virtual]
```

The deconstructor

Definition at line 467 of file Controller.cpp.

6.21.4 Member Function Documentation**6.21.4.1 getADC()**

```
virtual protocol::adc::ptr oCpt::iController::getADC (
    uint8_t id,
    uint8_t device ) [pure virtual]
```

A pure virtual function which gets a Pointer to a specific ADC

Parameters

<i>id</i>	The pin ID
<i>device</i>	the device ID

Returns

a pointer to the requested ADC

Implemented in [oCpt::ARM](#).

6.21.4.2 getAdcVector()

```
virtual std::vector<protocol::adc::ptr>* oCpt::iController::getAdcVector ( ) [pure virtual]
```

A pure virtual function which gets a pointer to all available ADC, if present. TODO check how it handles no ADC presents

Returns

a vector with pointers the all available ADCs

Implemented in [oCpt::ARM](#).

6.21.5 Member Data Documentation

6.21.5.1 adcVector_

`std::vector<protocol::adc::ptr> oCpt::iController::adcVector_ [protected]`

Definition at line 590 of file Controller.h.

Referenced by `oCpt::components::controller::BBB::BBB()`, `oCpt::ARM::getADC()`, and `oCpt::ARM::getAdcVector()`.

6.21.5.2 world_

`World::ptr oCpt::iController::world_ [protected]`

Definition at line 591 of file Controller.h.

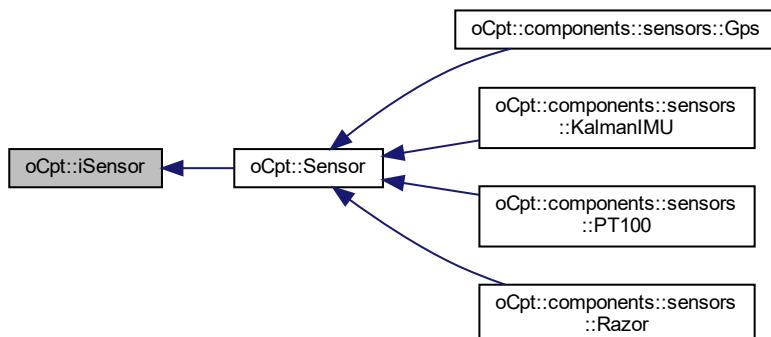
The documentation for this class was generated from the following files:

- /projects/mti/ohCaptain/ohCaptain/include/Core/[Controller.h](#)
- /projects/mti/ohCaptain/ohCaptain/src/Core/[Controller.cpp](#)

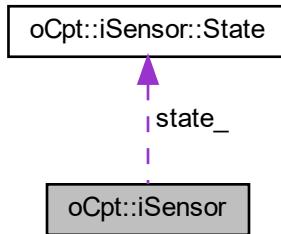
6.22 oCpt::iSensor Class Reference

#include <Sensor.h>

Inheritance diagram for oCpt::iSensor:



Collaboration diagram for oCpt::iSensor:



Classes

- struct [State](#)

Public Types

- typedef boost::shared_ptr< [iSensor](#) > ptr
- typedef boost::signals2::signal< void()> [signal_t](#)
- typedef boost::any [generic_t](#)

Public Member Functions

- [iSensor](#) ([iController::ptr](#) controller, [World::ptr](#) world, std::string id, std::string typeOfSensor="")
- virtual [~iSensor](#) ()
- virtual void [updateSensor](#) ()=0
- virtual void [run](#) ()=0
- virtual void [stop](#) ()=0
- virtual void [init](#) ()=0
- virtual void [setIoservice](#) (boost::shared_ptr< boost::asio::io_service > ioservice)=0
- virtual bool [operator==](#) ([iSensor::ptr](#) rhs)
- const boost::posix_time::milliseconds & [getTimer](#) () const
- void [setTimer](#) (const boost::posix_time::milliseconds &timer)
- [signal_t](#) & [getSig](#) ()
- const [State](#) & [getState](#) () const
- const std::string & [getID](#) () const
- void [setID](#) (const std::string &id)
- const std::string & [getTypeOfSensor](#) () const
- void [setTypeOfSensor](#) (const std::string &typeOfSensor)

Protected Attributes

- std::string `id_`
- std::string `typeOfSensor_`
- `iController::ptr controller_`
- `World::ptr world_`
- boost::posix_time::milliseconds `timer_`
- `signal_t sig_`
- `State state_`
- bool `sensorRunning_`
- boost::shared_ptr< boost::asio::io_service > `ioservice_`

6.22.1 Detailed Description

Each sensor that is used should adhere to the sensor interface. A sensor consists of a connection to a controller, such as a [ARM](#) device and the world. The sensor needs to be initiated with the construct, where afterwards the init function is called. The sensor should then be registered by the [Boatswain](#), using [Boatswain::registerSensor\(\)](#). This ensures that the boatswain can run the sensors. Some sensors are automatically update, whilst other need a manual action, such it is common practice to call the [iSensor::updateSensor\(\)](#). Once the value is update, a new Boost::Signal2 is fired, which allow for the main function to obtain the [State](#) of the sensor. Via [iSensor::getState\(\)](#). Since the return value of a sensor can vary, it is important to note that the final sensor should include a typedef with the return type named `ReturnValue_t`. After a sensor update is given or a signal is received, the return value can be CAST using the macro [CAST\(x,t\)](#)

Definition at line 29 of file Sensor.h.

6.22.2 Member Typedef Documentation

6.22.2.1 generic_t

```
typedef boost::any oCpt::iSensor::generic_t
```

Definition at line 33 of file Sensor.h.

6.22.2.2 ptr

```
typedef boost::shared_ptr<iSensor> oCpt::iSensor::ptr
```

Definition at line 31 of file Sensor.h.

6.22.2.3 signal_t

```
typedef boost::signals2::signal<void()> oCpt::iSensor::signal_t
```

Definition at line 32 of file Sensor.h.

6.22.3 Constructor & Destructor Documentation

6.22.3.1 iSensor()

```
oCpt::iSensor::iSensor (
    iController::ptr controller,
    World::ptr world,
    std::string id,
    std::string typeOfSensor = "" )
```

Constructor of [iSensor](#)

Parameters

<i>controller</i>	a shared_ptr of the controller where the sensor is hooked to
<i>world</i>	a shared_ptr of the world in which the vessel operates
<i>id</i>	a identifying name of the sensor
<i>typeOfSensor</i>	a identifying category for the sensor

Definition at line 10 of file Sensor.cpp.

References state_, and oCpt::iSensor::State::Value.

6.22.3.2 ~iSensor()

```
oCpt::iSensor::~iSensor ( ) [virtual]
```

Deconstructor of the sensor

Definition at line 18 of file Sensor.cpp.

6.22.4 Member Function Documentation**6.22.4.1 getID()**

```
const std::string & oCpt::iSensor::getID ( ) const
```

get the current ID

Returns

returns the ID as string

Definition at line 41 of file Sensor.cpp.

References id_.

6.22.4.2 getSig()

```
iSensor::signal_t & oCpt::iSensor::getSig ( )
```

get the signal that is to be fired when the state is updated

Returns

the signal_t

Definition at line 28 of file Sensor.cpp.

References sig_.

6.22.4.3 getState()

```
const iSensor::State & oCpt::iSensor::getState ( ) const
```

gets the last [State](#) of the sensor

Returns

the [State](#) object. Remember to CAST the value like such <sensorClass>::ReturnValue_t ret = CA←
ST(<sensorname>->[getState\(\)](#).Value, <sensorClass>);

Definition at line 32 of file Sensor.cpp.

References state_.

6.22.4.4 getTimer()

```
const boost::posix_time::milliseconds & oCpt::iSensor::getTimer ( ) const
```

Get the number of milliseconds when this sensor should be updated

Returns

returns a boost::posix_time::milliseconds type

Definition at line 20 of file Sensor.cpp.

References timer_.

6.22.4.5 getTypeOfSensor()

```
const std::string & oCpt::iSensor::getTypeOfSensor ( ) const
```

get the type of sensor

Returns

category identifying string

Definition at line 49 of file Sensor.cpp.

References typeOfSensor_.

6.22.4.6 init()

```
virtual void oCpt::iSensor::init ( ) [pure virtual]
```

pure virtual function for initializing the sensor

Implemented in [oCpt::components::sensors::KalmanIMU](#), [oCpt::Sensor](#), [oCpt::components::sensors::Razor](#), and [oCpt::components::sensors::PT100](#).

6.22.4.7 operator==()

```
bool oCpt::iSensor::operator== (
    iSensor::ptr rhs ) [virtual]
```

Equal operator determining if this sensor is equal with the pointer

Parameters

<i>rhs</i>	shared_ptr with the other sensor
------------	----------------------------------

Returns

returns either true or false

Definition at line 36 of file Sensor.cpp.

References controller_, id_, and typeOfSensor_.

6.22.4.8 run()

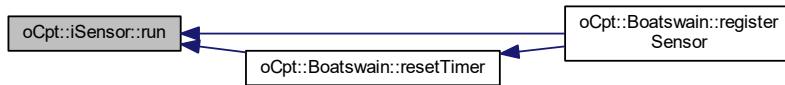
```
virtual void oCpt::iSensor::run ( ) [pure virtual]
```

pure virtual function for running of the sensor

Implemented in [oCpt::components::sensors::KalmanIMU](#), [oCpt::Sensor](#), [oCpt::components::sensors::Razor](#), [oCpt::components::sensors::PT100](#), and [oCpt::components::sensors::Gps](#).

Referenced by [oCpt::Boatswain::registerSensor\(\)](#), and [oCpt::Boatswain::resetTimer\(\)](#).

Here is the caller graph for this function:

**6.22.4.9 setID()**

```
void oCpt::iSensor::setID (
    const std::string & id )
```

sets the ID of the sensor

Parameters

<i>id</i>	identifying string
-----------	--------------------

Definition at line 45 of file Sensor.cpp.

References id_.

6.22.4.10 setIoservice()

```
virtual void oCpt::iSensor::setIoservice (
    boost::shared_ptr< boost::asio::io_service > ioservice ) [pure virtual]
```

pure virtual function for registering the Input Output service

Parameters

<i>ioservice</i>	the Input Output service used by Boost ASIO
------------------	---

Implemented in [oCpt::components::sensors::KalmanIMU](#), [oCpt::Sensor](#), [oCpt::components::sensors::Razor](#), and [oCpt::components::sensors::Gps](#).

6.22.4.11 setTimer()

```
void oCpt::iSensor::setTimer (
    const boost::posix_time::milliseconds & timer )
```

set the number of milliseconds when this sensor should be updated

Parameters

<i>timer</i>	the number of milliseconds as an boost::posix_time::milliseconds type
--------------	---

Definition at line 24 of file Sensor.cpp.

References `timer_`.

6.22.4.12 setTypeOfSensor()

```
void oCpt::iSensor::setTypeOfSensor (
    const std::string & typeOfSensor )
```

sets the category of the sensor, suchs as GPS, temperature

Parameters

<i>typeOfSensor</i>	category identifying string
---------------------	-----------------------------

Definition at line 53 of file Sensor.cpp.

References `typeOfSensor_`.

6.22.4.13 stop()

```
virtual void oCpt::iSensor::stop ( ) [pure virtual]
```

pure virtual function for stopping the sensor

Implemented in [oCpt::Sensor](#), [oCpt::components::sensors::Razor](#), [oCpt::components::sensors::PT100](#), and [oCpt::components::sensors::Gps](#).

6.22.4.14 updateSensor()

```
virtual void oCpt::iSensor::updateSensor ( ) [pure virtual]
```

pure virtual function for the updating of a sensor

Implemented in [oCpt::components::sensors::KalmanIMU](#), [oCpt::Sensor](#), [oCpt::components::sensors::Razor](#), [oCpt::components::sensors::PT100](#), and [oCpt::components::sensors::Gps](#).

6.22.5 Member Data Documentation

6.22.5.1 controller_

```
iController::ptr oCpt::iSensor::controller_ [protected]
```

Definition at line 140 of file Sensor.h.

Referenced by [operator==\(\)](#), and [oCpt::components::sensors::PT100::updateSensor\(\)](#).

6.22.5.2 id_

```
std::string oCpt::iSensor::id_ [protected]
```

Definition at line 112 of file Sensor.h.

Referenced by [getID\(\)](#), [operator==\(\)](#), and [setID\(\)](#).

6.22.5.3 ioservice_

```
boost::shared_ptr<boost::asio::io_service> oCpt::iSensor::ioservice_ [protected]
```

Definition at line 146 of file Sensor.h.

Referenced by [oCpt::Sensor::setIoservice\(\)](#).

6.22.5.4 sensorRunning_

```
bool oCpt::iSensor::sensorRunning_ [protected]
```

Definition at line 145 of file Sensor.h.

Referenced by [oCpt::components::sensors::Gps::run\(\)](#), [oCpt::components::sensors::Razor::run\(\)](#), [oCpt::Sensor::run\(\)](#), [oCpt::components::sensors::Gps::stop\(\)](#), [oCpt::components::sensors::Razor::stop\(\)](#), and [oCpt::Sensor::stop\(\)](#).

6.22.5.5 sig_

```
signal_t oCpt::iSensor::sig_ [protected]
```

Definition at line 143 of file Sensor.h.

Referenced by getSig(), oCpt::components::sensors::Gps::interpretMsg(), oCpt::components::sensors::Razor::msgHandler(), and oCpt::components::sensors::PT100::run().

6.22.5.6 state_

```
State oCpt::iSensor::state_ [protected]
```

Definition at line 144 of file Sensor.h.

Referenced by getState(), oCpt::components::sensors::Gps::interpretMsg(), iSensor(), oCpt::components::sensors::Razor::msgHandler(), oCpt::components::sensors::Razor::Razor(), oCpt::components::sensors::PT100::updateSensor(), and oCpt::components::sensors::Razor::updateSensor().

6.22.5.7 timer_

```
boost::posix_time::milliseconds oCpt::iSensor::timer_ [protected]
```

Definition at line 142 of file Sensor.h.

Referenced by getTimer(), and setTimer().

6.22.5.8 typeOfSensor_

```
std::string oCpt::iSensor::typeOfSensor_ [protected]
```

Definition at line 139 of file Sensor.h.

Referenced by getTypeOfSensor(), operator==(), and setTypeOfSensor().

6.22.5.9 world_

```
World::ptr oCpt::iSensor::world_ [protected]
```

Definition at line 141 of file Sensor.h.

Referenced by oCpt::components::sensors::Gps::interpretMsg(), oCpt::components::sensors::Razor::msgHandler(), oCpt::components::sensors::Razor::Razor(), oCpt::components::sensors::PT100::updateSensor(), and oCpt::components::sensors::Razor::updateSensor().

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

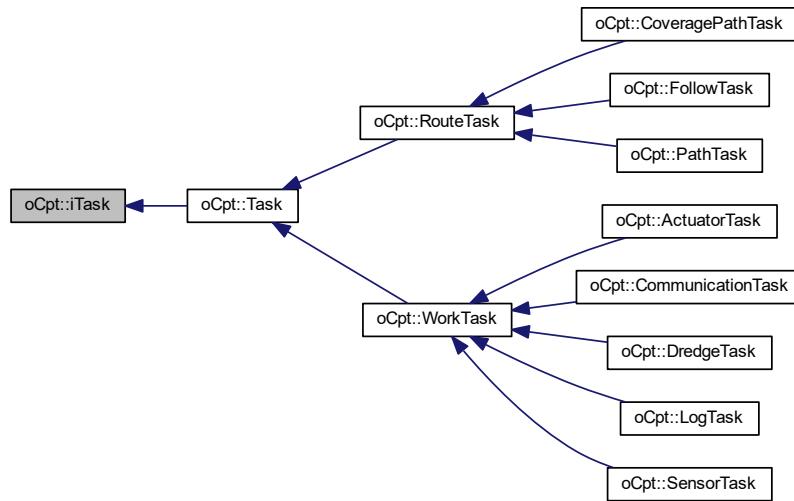
- /projects/mti/ohCaptain/ohCaptain/include/Core/[Sensor.h](#)
- /projects/mti/ohCaptain/ohCaptain/src/Core/[Sensor.cpp](#)

6.23 oCpt::iTask Class Reference

[Task](#) interface, all tasks need to adhere to this structure.

```
#include <Task.h>
```

Inheritance diagram for oCpt::iTask:



Classes

- class [Status](#)

Public Types

- enum [TypeOf](#) { **ROUTE** = 1, **WORK** = 2 }
- typedef boost::shared_ptr<[iTask](#)> **ptr**
Boost shared_ptr to a task.
- typedef std::list<[iTask::ptr](#)> **taskqueue**
A list of shared pointer tasks.

Public Member Functions

- [iTask \(iVessel::ptr vessel, bool concurrent=false\)](#)
- virtual [~iTask \(\)](#)
- virtual void [start \(\)=0](#)
- virtual [iTask::Status::ptr status \(\)=0](#)
- virtual void [stop \(\)=0](#)

Public Attributes

- [taskqueue Work](#)

Protected Attributes

- bool `_concurrent` = false
Allowed to run as a separate thread.
- `Vessel::ptr _vessel` = nullptr
Pointer to the world.

6.23.1 Detailed Description

`Task` interface, all tasks need to adhere to this structure.

This interface make sure that all task adheres to the same runtime rules and enable run-time polymorphism

Definition at line 20 of file Task.h.

6.23.2 Member Typedef Documentation

6.23.2.1 `ptr`

```
typedef boost::shared_ptr<iTask> oCpt::iTask::ptr
```

Boost shared_ptr to a task.

Definition at line 22 of file Task.h.

6.23.2.2 `taskqueue`

```
typedef std::list<iTask::ptr> oCpt::iTask::taskqueue
```

A list of shared pointer tasks.

Definition at line 23 of file Task.h.

6.23.3 Member Enumeration Documentation

6.23.3.1 `TypeOf`

```
enum oCpt::iTask::TypeOf
```

Enumeration indicating which type of task the object is

Enumerator

ROUTE	
WORK	

Definition at line 72 of file Task.h.

6.23.4 Constructor & Destructor Documentation

6.23.4.1 iTask()

```
oCpt::iTask::iTask (
    iVessel::ptr vessel,
    bool concurrent = false )
```

Constructor of the interface

Returns

Definition at line 8 of file Task.cpp.

References `_concurrent`, and `_vessel`.

6.23.4.2 ~iTask()

```
oCpt::iTask::~iTask ( ) [virtual]
```

Deconstructor of the interface

Definition at line 13 of file Task.cpp.

6.23.5 Member Function Documentation

6.23.5.1 start()

```
virtual void oCpt::iTask::start ( ) [pure virtual]
```

The start command for a task

Implemented in [oCpt::Task](#).

6.23.5.2 status()

```
virtual iTASK::Status::ptr oCpt::iTask::status ( ) [pure virtual]
```

Retrieves the [Status](#) of a task

Returns

Boost shared_ptr of the task status

Implemented in [oCpt::Task](#).

6.23.5.3 stop()

```
virtual void oCpt::iTask::stop ( ) [pure virtual]
```

The stop command for a task

Implemented in [oCpt::Task](#).

6.23.6 Member Data Documentation

6.23.6.1 _concurrent

```
bool oCpt::iTask::_concurrent = false [protected]
```

Allowed to run as a seperate thread.

Definition at line 104 of file Task.h.

Referenced by [iTTask\(\)](#).

6.23.6.2 _vessel

```
Vessel::ptr oCpt::iTask::_vessel = nullptr [protected]
```

Pointer to the world.

Definition at line 105 of file Task.h.

Referenced by [iTTask\(\)](#).

6.23.6.3 Work

```
taskqueue oCpt::iTask::Work
```

Definition at line 25 of file Task.h.

Referenced by [oCpt::Task::start\(\)](#).

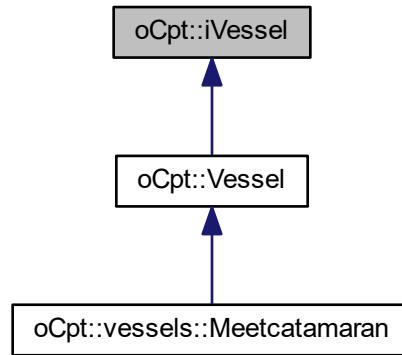
The documentation for this class was generated from the following files:

- /projects/mti/ohCaptain/ohCaptain/include/Core/[Task.h](#)
- /projects/mti/ohCaptain/ohCaptain/src/Core/[Task.cpp](#)

6.24 oCpt::iVessel Class Reference

```
#include <Vessel.h>
```

Inheritance diagram for oCpt::iVessel:



Public Types

- `typedef boost::shared_ptr< iVessel > ptr`
Boost shared_ptr to a vessel.

Public Member Functions

- `iVessel ()`
- `iVessel (iController::ptr controller)`
- `virtual ~iVessel ()`
- `virtual void initialize ()=0`
- `virtual void run ()=0`
- `virtual void stop ()=0`
- `const boost::shared_ptr< bool > & getStopThread () const`
- `void setStopThread (const boost::shared_ptr< bool > &stopThread)`

Protected Attributes

- `boost::shared_ptr< bool > stopThread_`
The global shared pointer for stopping all threads.

6.24.1 Detailed Description

The interface for each vessel

Definition at line 24 of file Vessel.h.

6.24.2 Member Typedef Documentation

6.24.2.1 ptr

```
typedef boost::shared_ptr<iVessel> oCpt::iVessel::ptr
```

Boost shared_ptr to a vessel.

Definition at line 26 of file Vessel.h.

6.24.3 Constructor & Destructor Documentation

6.24.3.1 iVessel() [1/2]

```
oCpt::iVessel::iVessel ( )
```

Constructor of the vessel interface

Returns

Definition at line 9 of file Vessel.cpp.

6.24.3.2 iVessel() [2/2]

```
oCpt::iVessel::iVessel (
    iController::ptr controller )
```

Constructor of the vessel interface

Parameters

<i>controller</i>	shared_ptr to the controller
-------------------	------------------------------

Returns

Definition at line 11 of file Vessel.cpp.

6.24.3.3 ~iVessel()

```
oCpt::iVessel::~iVessel ( ) [virtual]
```

Deconstructor

Definition at line 13 of file Vessel.cpp.

6.24.4 Member Function Documentation

6.24.4.1 getStopThread()

```
const boost::shared_ptr< bool > & oCpt::iVessel::getStopThread ( ) const
```

Get the stop thread variable

Returns

shared_ptr for each each thread;

Definition at line 15 of file Vessel.cpp.

References stopThread_.

6.24.4.2 initialize()

```
virtual void oCpt::iVessel::initialize ( ) [pure virtual]
```

Initialize the vessel

Implemented in [oCpt::Vessel](#).

6.24.4.3 run()

```
virtual void oCpt::iVessel::run ( ) [pure virtual]
```

Run the vessel normal operations

Implemented in [oCpt::Vessel](#).

6.24.4.4 setStopThread()

```
void oCpt::iVessel::setStopThread (   
    const boost::shared_ptr< bool > & stopThread )
```

Set the stop thread variable

Parameters

<code>stopThread</code>	a shared_ptr for all threads
-------------------------	------------------------------

Definition at line 19 of file Vessel.cpp.

References stopThread_.

6.24.4.5 stop()

```
virtual void oCpt::iVessel::stop () [pure virtual]
```

Stop the vessel, everything except critical parts, which are needed to survive

Implemented in [oCpt::Vessel](#).

6.24.5 Member Data Documentation

6.24.5.1 stopThread_

```
boost::shared_ptr<bool> oCpt::iVessel::stopThread_ [protected]
```

The global shared pointer for stopping all threads.

Definition at line 74 of file [Vessel.h](#).

Referenced by [getStopThread\(\)](#), [setStopThread\(\)](#), [oCpt::Vessel::stop\(\)](#), and [oCpt::Vessel::Vessel\(\)](#).

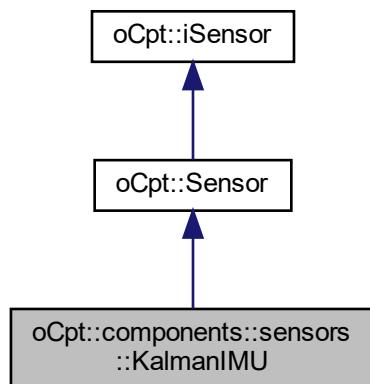
The documentation for this class was generated from the following files:

- /projects/mti/ohCaptain/ohCaptain/include/Core/[Vessel.h](#)
- /projects/mti/ohCaptain/ohCaptain/src/Core/[Vessel.cpp](#)

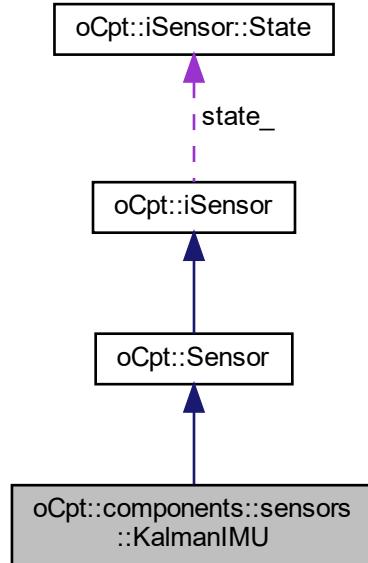
6.25 oCpt::components::sensors::KalmanIMU Class Reference

```
#include <KalmanIMU.h>
```

Inheritance diagram for oCpt::components::sensors::KalmanIMU:



Collaboration diagram for oCpt::components::sensors::KalmanIMU:



Classes

- struct [ReturnValue](#)

Public Types

- `typedef struct oCpt::components::sensors::KalmanIMU::ReturnValue ReturnValue_t`

Public Member Functions

- `KalmanIMU (iController::ptr controller, World::ptr world, std::string id, Razor::ptr RazorSensor)`
- `~KalmanIMU ()`
- `void updateSensor ()`
- `void run ()`
- `void init ()`
- `void setIoservice (boost::shared_ptr< boost::asio::io_service > ioservice)`

Private Member Functions

- `void RazorUpdate ()`

Private Attributes

- `std::string device_`
- `Razor::ptr razor_`

Additional Inherited Members

6.25.1 Detailed Description

Definition at line 229 of file KalmanIMU.h.

6.25.2 Member Typedef Documentation

6.25.2.1 ReturnValue_t

```
typedef struct oCpt::components::sensors::KalmanIMU::ReturnValue oCpt::components::sensors::←  
KalmanIMU::ReturnValue_t
```

6.25.3 Constructor & Destructor Documentation

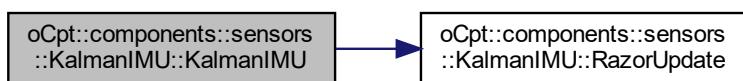
6.25.3.1 KalmanIMU()

```
oCpt::components::sensors::KalmanIMU::KalmanIMU (   
    iController::ptr controller,  
    World::ptr world,  
    std::string id,  
    Razor::ptr RazorSensor )
```

Definition at line 7 of file KalmanIMU.cpp.

References razor_, and RazorUpdate().

Here is the call graph for this function:



6.25.3.2 ~KalmanIMU()

```
oCpt::components::sensors::KalmanIMU::~KalmanIMU ( )
```

Definition at line 14 of file KalmanIMU.cpp.

6.25.4 Member Function Documentation

6.25.4.1 init()

```
void oCpt::components::sensors::KalmanIMU::init ( ) [virtual]
```

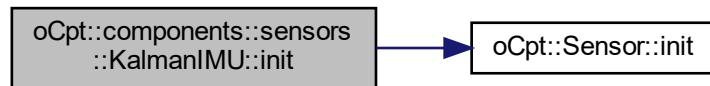
Initialize the sensor

Reimplemented from [oCpt::Sensor](#).

Definition at line 26 of file KalmanIMU.cpp.

References [oCpt::Sensor::init\(\)](#).

Here is the call graph for this function:



6.25.4.2 RazorUpdate()

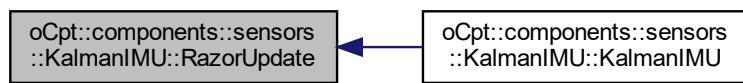
```
void oCpt::components::sensors::KalmanIMU::RazorUpdate ( ) [private]
```

Definition at line 34 of file KalmanIMU.cpp.

References [oCpt::components::sensors::Razor::ReturnValue::acc](#), [CAST](#), and [razor_](#).

Referenced by [KalmanIMU\(\)](#).

Here is the caller graph for this function:



6.25.4.3 run()

```
void oCpt::components::sensors::KalmanIMU::run ( ) [virtual]
```

virtual function starting the run service for the IO

Reimplemented from [oCpt::Sensor](#).

Definition at line 22 of file KalmanIMU.cpp.

References [oCpt::Sensor::run\(\)](#).

Here is the call graph for this function:



6.25.4.4 setIoservice()

```
void oCpt::components::sensors::KalmanIMU::setIoservice (
    boost::shared_ptr< boost::asio::io_service > ioservice ) [virtual]
```

Setting the used Asynchronous Input Output service

Parameters

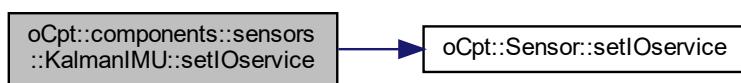
<i>ioservice</i>	ASIO IO service, which handles the async calls from multiple sensors
------------------	--

Reimplemented from [oCpt::Sensor](#).

Definition at line 30 of file KalmanIMU.cpp.

References [oCpt::Sensor::setIoservice\(\)](#).

Here is the call graph for this function:



6.25.4.5 updateSensor()

```
void oCpt::components::sensors::KalmanIMU::updateSensor ( ) [virtual]
```

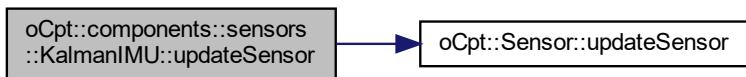
virtual function which performs a sensor update, obtaining a new value and sending a signal afterwards

Reimplemented from [oCpt::Sensor](#).

Definition at line 18 of file KalmanIMU.cpp.

References [oCpt::Sensor::updateSensor\(\)](#).

Here is the call graph for this function:



6.25.5 Member Data Documentation

6.25.5.1 device_

```
std::string oCpt::components::sensors::KalmanIMU::device_ [private]
```

Definition at line 252 of file KalmanIMU.h.

6.25.5.2 razor_

```
Razor::ptr oCpt::components::sensors::KalmanIMU::razor_ [private]
```

Definition at line 253 of file KalmanIMU.h.

Referenced by [KalmanIMU\(\)](#), and [RazorUpdate\(\)](#).

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

- /projects/mti/ohCaptain/ohCaptain/include/Sensors/[KalmanIMU.h](#)
- /projects/mti/ohCaptain/ohCaptain/src/Sensors/[KalmanIMU.cpp](#)

6.26 oCpt::World::Location Class Reference

```
#include <World.h>
```

Classes

- struct `coordinate`
- struct `gpsPoint`
- struct `RoutePoint`

Public Types

- enum `cardinal_direction` { `North` = 110, `South` = 115, `East` = 101, `West` = 119 }
- typedef `quantity<boost::units::degree::plane_angle, double>` `degree_t`
- typedef struct `oCpt::World::Location::coordinate` `coordinate_t`
- typedef struct `oCpt::World::Location::gpsPoint` `gpsPoint_t`
- typedef `boost::shared_ptr<Location>` `ptr`

Public Member Functions

- `Location()`
- virtual `~Location()`
- `RoutePoint::ptr getCurrentLocation(bool newMeasurement=false)`
- void `push_back(RoutePoint::ptr routePoint)`
- `std::vector<RoutePoint::ptr> getLocationHistory()`

Static Public Member Functions

- static `cardinal_direction stocd(std::string str)`

Private Attributes

- `RoutePoint::ptr currentLocation_`
- `std::vector<RoutePoint::ptr> LocationHistory`

6.26.1 Detailed Description

A location in the `World`

Definition at line 123 of file `World.h`.

6.26.2 Member Typedef Documentation

6.26.2.1 coordinate_t

```
typedef struct oCpt::World::Location::coordinate oCpt::World::Location::coordinate_t
```

6.26.2.2 degree_t

```
typedef quantity<boost::units::degree::plane_angle, double> oCpt::World::Location::degree_t
```

Definition at line 125 of file `World.h`.

6.26.2.3 gpsPoint_t

```
typedef struct oCpt::World::Location::gpsPoint oCpt::World::Location::gpsPoint_t
```

6.26.2.4 ptr

```
typedef boost::shared_ptr<Location> oCpt::World::Location::ptr
```

Definition at line 151 of file World.h.

6.26.3 Member Enumeration Documentation

6.26.3.1 cardinal_direction

```
enum oCpt::World::Location::cardinal_direction
```

Enumerator

North	enum value North
South	enum value South
East	enum value East
West	enum value West

Definition at line 127 of file World.h.

6.26.4 Constructor & Destructor Documentation

6.26.4.1 Location()

```
oCpt::World::Location::Location ( )
```

Constructor for the [Location](#)

Definition at line 37 of file World.cpp.

6.26.4.2 ~Location()

```
oCpt::World::Location::~Location ( ) [virtual]
```

Deconstruction for the [Location](#)

Definition at line 41 of file World.cpp.

6.26.5 Member Function Documentation

6.26.5.1 getCurrentLocation()

```
World::Location::RoutePoint::ptr oCpt::World::Location::getCurrentLocation (
    bool newMeasurement = false )
```

get the current [Location](#)

Parameters

<i>newMeasurement</i>	should a new measurement be executed? or is the latest log sufficient
-----------------------	---

Returns

returns the last Way point

Definition at line 45 of file World.cpp.

6.26.5.2 getlocationHistory()

```
std::vector< World::Location::RoutePoint::ptr > oCpt::World::Location::getlocationHistory ( )
```

Get the complete location history

Returns

returns a vector with shared_ptr of all waypoints reached

Definition at line 53 of file World.cpp.

6.26.5.3 push_back()

```
void oCpt::World::Location::push_back (
    RoutePoint::ptr routePoint )
```

Add a new waypoint to the history log

Parameters

<i>routePoint</i>	a waypoint
-------------------	------------

Definition at line 49 of file World.cpp.

6.26.5.4 stocd()

```
World::Location::cardinal_direction oCpt::World::Location::stocd (
    std::string str ) [static]
```

Convert a string to a cardinal direction

Parameters

<i>str</i>	North/north/N/n / West,west,W,w / South,south,S,s / East,east,E,e are taken as argument
------------	---

Returns

a cardinal direction

Definition at line 57 of file World.cpp.

Referenced by oCpt::components::sensors::Gps::interpretMsg().

Here is the caller graph for this function:



6.26.6 Member Data Documentation

6.26.6.1 currentLocation_

`RoutePoint::ptr oCpt::World::Location::currentLocation_ [private]`

Definition at line 196 of file World.h.

6.26.6.2 LocationHistory

`std::vector<RoutePoint::ptr> oCpt::World::Location::LocationHistory [private]`

Definition at line 197 of file World.h.

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

- /projects/mti/ohCaptain/ohCaptain/include/Core/[World.h](#)
- /projects/mti/ohCaptain/ohCaptain/src/Core/[World.cpp](#)

6.27 oCpt::World::Time::Log< T > Class Template Reference

#include <World.h>

Public Types

- `typedef boost::shared_ptr< Log > ptr`

Public Member Functions

- [Log \(\)](#)
- [Log \(const T &value, const timepoint_t &epoch=clock_t::now\(\)\)](#)
- virtual [~Log \(\)](#)
- const [timepoint_t & getEpoch \(\) const](#)
- const T & [getValue \(\) const](#)

Private Attributes

- [timepoint_t _epoch](#)
- [T _value](#)

6.27.1 Detailed Description

```
template<typename T>
class oCpt::World::Time::Log< T >
```

A template class to [Log](#) generic values at an certain epoch in time

Template Parameters

<i>T</i>	Type of value to log
----------	----------------------

Definition at line 59 of file World.h.

6.27.2 Member Typedef Documentation

6.27.2.1 ptr

```
template<typename T >
typedef boost::shared_ptr<Log> oCpt::World::Time::Log< T >::ptr
```

Definition at line 61 of file World.h.

6.27.3 Constructor & Destructor Documentation

6.27.3.1 Log() [1/2]

```
template<typename T >
oCpt::World::Time::Log< T >::Log ( ) [inline]
```

Constructor of the [Log](#) class

Definition at line 70 of file World.h.

6.27.3.2 Log() [2/2]

```
template<typename T >
oCpt::World::Time::Log< T >::Log (
    const T & value,
    const timepoint_t & epoch = clock_t::now() ) [inline]
```

Constructor of the [Log](#) class

Parameters

<i>value</i>	The Value to store
<i>epoch</i>	the Time point, with a default to the now moment

Definition at line 77 of file World.h.

6.27.3.3 ~Log()

```
template<typename T >
virtual oCpt::World::Time::Log< T >::~Log ( ) [inline], [virtual]
```

Deconstructor of the [Log](#) class

Definition at line 85 of file World.h.

6.27.4 Member Function Documentation**6.27.4.1 getEpoch()**

```
template<typename T >
const timepoint\_t& oCpt::World::Time::Log< T >::getEpoch ( ) const [inline]
```

Get the current Epoch

Returns

returns a time point when the [Log](#) has taken place

Definition at line 91 of file World.h.

6.27.4.2 getValue()

```
template<typename T >
const T& oCpt::World::Time::Log< T >::getValue ( ) const [inline]
```

Gets the current value

Returns

returns the value at an certain time

Definition at line 99 of file World.h.

6.27.5 Member Data Documentation

6.27.5.1 `_epoch`

```
template<typename T >
timepoint_t oCpt::World::Time::Log< T >::_epoch [private]
```

Definition at line 63 of file World.h.

6.27.5.2 `_value`

```
template<typename T >
T oCpt::World::Time::Log< T >::_value [private]
```

Definition at line 64 of file World.h.

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

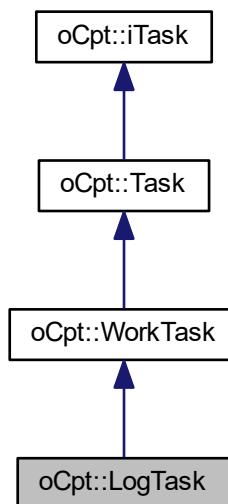
- /projects/mti/ohCaptain/ohCaptain/include/Core/World.h

6.28 oCpt::LogTask Class Reference

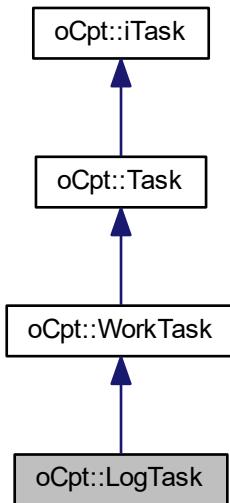
An Object representing a data logging task.

```
#include <Task.h>
```

Inheritance diagram for oCpt::LogTask:



Collaboration diagram for oCpt::LogTask:



Public Member Functions

- [LogTask \(`Vessel::ptr` vessel, `bool concurrent=true`\)](#)
- virtual [~LogTask \(\)](#)

Additional Inherited Members

6.28.1 Detailed Description

An Object representing a data logging task.

All these types of tasks make use of a sensor to record and log

Definition at line 265 of file Task.h.

6.28.2 Constructor & Destructor Documentation

6.28.2.1 LogTask()

```

oCpt::LogTask::LogTask (
    Vessel::ptr vessel,
    bool concurrent = true )
  
```

Constructor of the interface

Returns

Definition at line 65 of file Task.cpp.

6.28.2.2 ~LogTask()

```
oCpt::LogTask::~LogTask ( ) [virtual]
```

The deconstructor

Definition at line 67 of file Task.cpp.

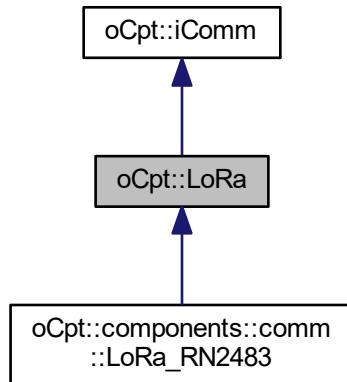
The documentation for this class was generated from the following files:

- /projects/mti/ohCaptain/ohCaptain/include/Core/[Task.h](#)
- /projects/mti/ohCaptain/ohCaptain/src/Core/[Task.cpp](#)

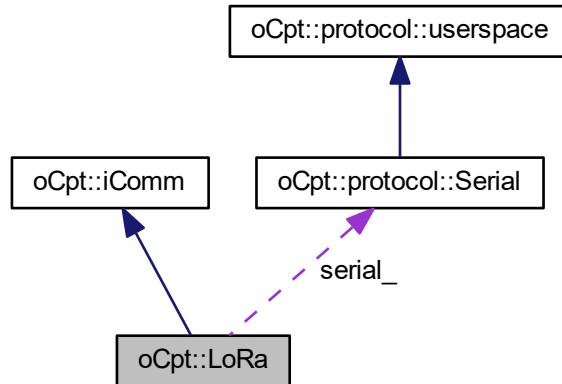
6.29 oCpt::LoRa Class Reference

```
#include <Communication.h>
```

Inheritance diagram for oCpt::LoRa:



Collaboration diagram for oCpt::LoRa:



Public Types

- enum `ModulationMode` { `LORA` = 1, `FSK` = 2 }
- enum `SpreadingFactor` {
 `SF7` = 7, `SF8` = 8, `SF9` = 9, `SF10` = 10,
 `SF11` = 11, `SF12` = 12 }
- enum `BandWidth` {
 `BW250` = 1, `BW200` = 2, `BW166_7` = 3, `BW125` = 4,
 `BW100` = 5, `BW83_3` = 6, `BW62_5` = 7, `BW50` = 8,
 `BW41_7` = 9, `BW31_3` = 10, `BW25` = 11, `BW20_8` = 12,
 `BW15_6` = 13, `BW12_5` = 14, `BW10_4` = 15, `BW7_8` = 16,
 `BW6_3` = 17, `BW5_2` = 18, `BW3_9` = 19, `BW3_1` = 20,
 `BW2_6` = 21 }
- enum `CodingRate` { `CR4_5` = 4, `CR4_6` = 3, `CR4_7` = 2, `CR4_8` = 1 }
- enum `RadioBandWidth` { `RBW500` = 500, `RBW250` = 250, `RBW125` = 125 }
- enum `GetSet` { `GET`, `SET`, `NONE` }
- enum `RadioCommand` {
 `MOD`, `FREQ`, `PWR`, `SF`,
 `AFCBW`, `RXBW`, `FSKBITRATE`, `FDEV`,
 `PRLEN`, `CRC`, `CR`, `WDT`,
 `SYNC`, `BW`, `rRX`, `rTX` }
- enum `MacCommand` {
 `PAUSE`, `RESET`, `mTX`, `JOIN`,
 `SAVE`, `FORCEENABLE`, `RESUME` }

Public Member Functions

- `LoRa` (const std::string &`id`, const std::string &`device`, `World::ptr` `world`=`World::ptr`(new `World()`), `iController`
 `::io_t` `ioservice`=`iController::io_t`(new boost::asio::io_service()))
- virtual ~`LoRa` ()
- virtual void `run` () override
- virtual void `stop` () override

- virtual void `initialize` () override
- virtual void `sendMessage` (Message msg) override
- virtual Message::ptr `recieveMessage` () override
- virtual void `recieveAsyncMessage` () override

Protected Member Functions

- void `messageReceived` ()
- std::string `bandWidthToString` (const BandWidth &value)
- std::string `codingRateToString` (const CodingRate &value)
- void `stringToHex` (const std::string str, std::string &hexStr, const bool capital=true)
- void `hexToString` (const std::string hexStr, std::string &str)
- template<typename T>
std::string `encodeTypeToHex` (T value, bool capital=true)
- template<typename T>
std::string `buildMacCmdString` (MacCommand cmd, T value=0, GetSet prop=NONE)
- template<typename T>
std::string `buildRadioCmdString` (RadioCommand cmd, T value=0, GetSet prop=SET)
- std::string `buildRadioCmdString` (RadioCommand cmd, std::string value, GetSet prop=SET)
- unsigned long `calculateDownTime` (unsigned int payload)
- void `write` (const std::string &value)
- void `rx` ()
- void `macpause` ()

Protected Attributes

- bool `proceed_`
- bool `ignoreWarn_`
- bool `listen_`
- protocol::Serial `serial_`
SerialPort for UART communication with the chip.
- unsigned int `baudrate_`
- ModulationMode `mod_`
Modulation mode.
- unsigned long `freq_`
Frequency between 433050000..4347900000 or 863000000...870000000.
- int8_t `pwr_`
Power of transmission between -3...15.
- SpreadingFactor `sf_`
Spreading factor of the signal.
- BandWidth `afcbw_`
Automatic frequency correction in kHz.
- BandWidth `rxbw_`
Signal bandwidth in kHz.
- uint `fskBitRate_`
FSK bitrate between 1...300000.
- uint `fdev_`
Frequency deviation between 0...200000.
- uint `plen_`
Preamble length between 0...65535.
- bool `crc_`

- `CodingRate cr_`
The coding rate.
- `unsigned long wdt_`
WatchDog 0...4294967295. Set to 0 to disable.
- `unsigned int sync_`
Sync word.
- `RadioBandWidth bw_`
RadioBandWidth in kHz.
- `bool sendAllowed_`

Additional Inherited Members

6.29.1 Detailed Description

Communication class for the [LoRa](#) protocol. The current class is mostly based on node 2 node communication
 TODO rewrite so it will allow mesh network communication. Most of the commands are taken from <http://ww1.microchip.com/downloads/en/DeviceDoc/40001784B.pdf>

Definition at line 160 of file Communication.h.

6.29.2 Member Enumeration Documentation

6.29.2.1 BandWidth

```
enum oCpt::LoRa::BandWidth
```

The bandwidth

Enumerator

BW250	
BW200	
BW166 ₇	
BW125	
BW100	
BW83_3	
BW62_5	
BW50	
BW41_7	
BW31_3	
BW25	
BW20_8	
BW15_6	
BW12_5	
BW10_4	
BW7_8	
BW6_3	
BW5_2	
BW3_9	
BW3_1	
BW2_6	

Definition at line 186 of file Communication.h.

6.29.2.2 CodingRate

```
enum oCpt::LoRa::CodingRate
```

The Coding rate of the signal

Enumerator

CR4↔_5	
CR4↔_6	
CR4↔_7	
CR4↔_8	

Definition at line 213 of file Communication.h.

6.29.2.3 GetSet

```
enum oCpt::LoRa::GetSet
```

Perform a get or a set command or otherwise none

Enumerator

GET	
SET	
NONE	

Definition at line 232 of file Communication.h.

6.29.2.4 MacCommand

```
enum oCpt::LoRa::MacCommand
```

Type to control the MAC layer, currently only pause is used, because node 2 node communication doesn't use MAC

Enumerator

PAUSE	
RESET	
mTX	
JOIN	
SAVE	
FORCEENABLE	
RESUME	

Definition at line 263 of file Communication.h.

6.29.2.5 ModulationMode

enum `oCpt::LoRa::ModulationMode`

The modulation mode of the [LoRa](#) module

Enumerator

LORA	
FSK	

Definition at line 166 of file Communication.h.

6.29.2.6 RadioBandWidth

enum `oCpt::LoRa::RadioBandWidth`

The radio bandwidth

Enumerator

RBW500	
RBW250	
RBW125	

Definition at line 223 of file Communication.h.

6.29.2.7 RadioCommand

enum `oCpt::LoRa::RadioCommand`

Types of radio commands

Enumerator

MOD	
FREQ	
PWR	
SF	
AFCBW	
RXBW	
FSKBITRATE	
FDEV	
PRLEN	
CRC	
CR	
WDT	

Enumerator

SYNC	
BW	
rRX	
rTX	

Definition at line 241 of file Communication.h.

6.29.2.8 SpreadingFactor

```
enum oCpt::LoRa::SpreadingFactor
```

The Spreading factor

Enumerator

SF7	
SF8	
SF9	
SF10	
SF11	
SF12	

Definition at line 174 of file Communication.h.

6.29.3 Constructor & Destructor Documentation

6.29.3.1 LoRa()

```
oCpt::LoRa::LoRa (
    const std::string & id,
    const std::string & device,
    World::ptr world = World::ptr(new World()),
    iController::io_t ioservice = iController::io_t(new boost::asio::io_service()) )
```

LoRa device constructor

Parameters

<i>id</i>	the ID of the device as an string
<i>device</i>	the device path eq. /dev/ttyS0
<i>world</i>	shared_ptr to the World default = a newly created World
<i>ioservice</i>	shared_ptr to an IO service, default is a newly created IO service

Definition at line 58 of file Communication.cpp.

6.29.3.2 ~LoRa()

```
oCpt::LoRa::~LoRa ( ) [virtual]
```

Definition at line 81 of file Communication.cpp.

6.29.4 Member Function Documentation

6.29.4.1 bandWidthToString()

```
std::string oCpt::LoRa::bandWidthToString (
    const BandWidth & value ) [protected]
```

Definition at line 101 of file Communication.cpp.

References BW100, BW10_4, BW125, BW12_5, BW15_6, BW166_7, BW200, BW20_8, BW25, BW250, BW31_3, BW3_1, BW3_9, BW41_7, BW50, BW5_2, BW62_5, BW6_3, BW7_8, and BW83_3.

6.29.4.2 buildMacCmdString()

```
template<typename T >
std::string oCpt::LoRa::buildMacCmdString (
    MacCommand cmd,
    T value = 0,
    GetSet prop = NONE ) [inline], [protected]
```

A command string builder for MAC commands currently onlu PAUSE implemented

Template Parameters

<i>T</i>	the type of MAC command eq. MacCommand::PAUSE
----------	---

Parameters

<i>cmd</i>	the MacCommand to be performed
<i>value</i>	the Value to be send
<i>prop</i>	Additional properties

Returns

a string which can be send to the LoRa module eq. "mac set pause"

Definition at line 339 of file Communication.h.

6.29.4.3 buildRadioCmdString() [1/2]

```
template<typename T >
std::string oCpt::LoRa::buildRadioCmdString (
```

```
RadioCommand cmd,
T value = 0,
GetSet prop = SET ) [inline], [protected]
```

A command string builder for radio commands

Template Parameters

T	
---	--

Parameters

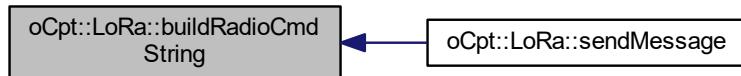
cmd	
value	
prop	

Returns

Definition at line 367 of file Communication.h.

Referenced by sendMessage().

Here is the caller graph for this function:



6.29.4.4 buildRadioCmdString() [2/2]

```
std::string oCpt::LoRa::buildRadioCmdString (
    RadioCommand cmd,
    std::string value,
    GetSet prop = SET ) [inline], [protected]
```

Definition at line 437 of file Communication.h.

6.29.4.5 calculateDownTime()

```
unsigned long oCpt::LoRa::calculateDownTime (
    unsigned int payload ) [protected]
```

Definition at line 85 of file Communication.cpp.

References bw_, cr_, crc_, prlen_, and sf_.

6.29.4.6 codingRateToString()

```
std::string oCpt::LoRa::codingRateToString (
    const CodingRate & value ) [protected]
```

Definition at line 148 of file Communication.cpp.

References CR4_5, CR4_6, and CR4_7.

6.29.4.7 encodeTypeToHex()

```
template<typename T >
std::string oCpt::LoRa::encodeTypeToHex (
    T value,
    bool capital = true ) [inline], [protected]
```

Convert the value of a Type T to a hexidecimal string, which can be send to a [LoRa](#) device, such that it can be transmitted

Template Parameters

<i>T</i>	the type of value, to be converted
----------	------------------------------------

Parameters

<i>value</i>	the to be converted value
<i>capital</i>	boolean indicating if the hexidecimal string should consist of capital letters

Returns

a string with the value as hexidecimal values

Definition at line 316 of file Communication.h.

References boost::units::constants::c.

6.29.4.8 hexToString()

```
void oCpt::LoRa::hexToString (
    const std::string hexStr,
    std::string & str ) [protected]
```

Definition at line 280 of file Communication.cpp.

Referenced by messageRecieved().

Here is the caller graph for this function:



6.29.4.9 initialize()

```
void oCpt::LoRa::initialize ( ) [override], [virtual]
```

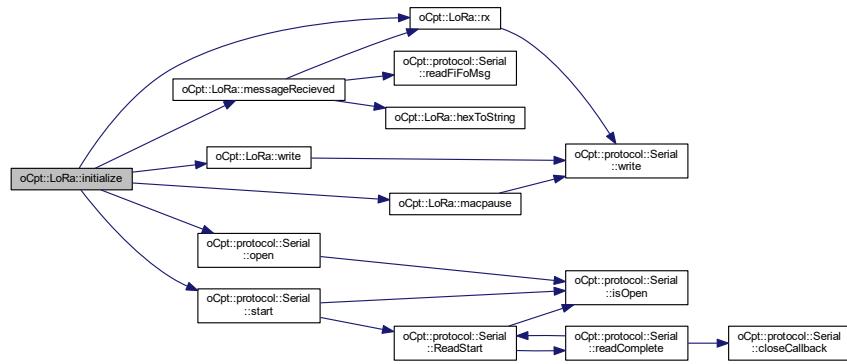
A pure virtual function which initializes the communication device

Implements [oCpt::iComm](#).

Definition at line 189 of file Communication.cpp.

References AFCBW, afcbw_, BW, bw_, CR, cr_, CRC, crc_, FDEV, fdev_, FREQ, freq_, ignoreWarn_, listen_←, macpause(), messageRecieved(), MOD, mod_, oCpt::protocol::Serial::msgRecievedSig, oCpt::protocol::Serial::open(), PRLEN, prlen_, PWR, pwr_, rx(), RXBW, rxbw_, serial_, SF, sf_, oCpt::protocol::Serial::start(), SYNC, sync_, WDT, wdt_, and write().

Here is the call graph for this function:



6.29.4.10 macpause()

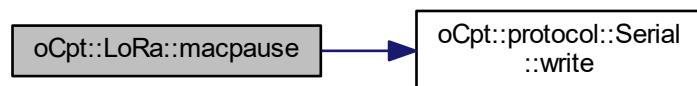
```
void oCpt::LoRa::macpause ( ) [protected]
```

Definition at line 275 of file Communication.cpp.

References PAUSE, serial_, and oCpt::protocol::Serial::write().

Referenced by `initialize()`.

Here is the call graph for this function:



Here is the caller graph for this function:



6.29.4.11 messageRecieved()

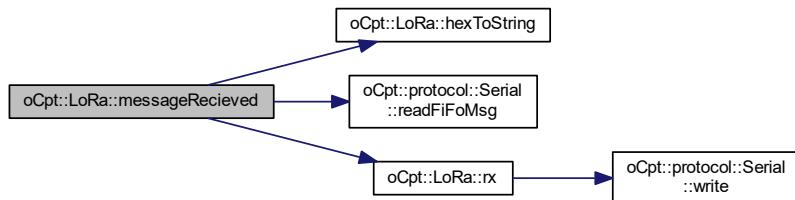
```
void oCpt::LoRa::messageRecieved() [protected]
```

Definition at line 161 of file Communication.cpp.

References hexToString(), ignoreWarn_, listen_, oCpt::iComm::msgQueue_, oCpt::iComm::msgRecievedSig, proceed_, oCpt::protocol::Serial::readFiFoMsg(), rx(), serial_, and oCpt::iComm::world_.

Referenced by initialize().

Here is the call graph for this function:



Here is the caller graph for this function:



6.29.4.12 receiveAsyncMessage()

```
void oCpt::LoRa::receiveAsyncMessage ( ) [override], [virtual]
```

A pure virtual function which performs the polling for a new message on a separate threads, so it won't block the current one, it needs to send a signal when the message is received

Implements [oCpt::iComm](#).

Definition at line 241 of file Communication.cpp.

6.29.4.13 recieveMessage()

```
iComm::Message::ptr oCpt::LoRa::recieveMessage ( ) [override], [virtual]
```

A pure virtual function with a shared_ptr to the first in queue received message, this function will hold the current thread

Returns

a shared_ptr pointing towards the queued Message

Implements [oCpt::iComm](#).

Definition at line 237 of file Communication.cpp.

6.29.4.14 run()

```
void oCpt::LoRa::run ( ) [override], [virtual]
```

a pure virtual function which runs the communication device

Implements [oCpt::iComm](#).

Definition at line 181 of file Communication.cpp.

6.29.4.15 rx()

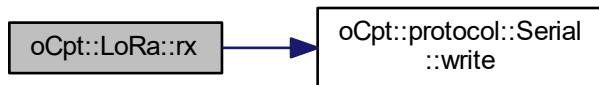
```
void oCpt::LoRa::rx ( ) [protected]
```

Definition at line 270 of file Communication.cpp.

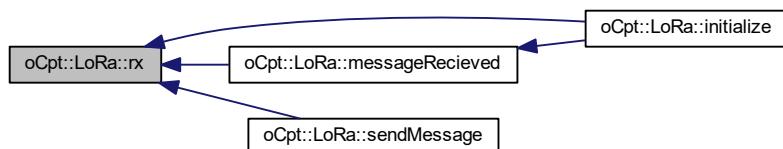
References NONE, rRX, serial_, and oCpt::protocol::Serial::write().

Referenced by initialize(), messageRecieved(), and sendMessage().

Here is the call graph for this function:



Here is the caller graph for this function:



6.29.4.16 sendMessage()

```
void oCpt::LoRa::sendMessage (
    Message msg ) [override], [virtual]
```

A pure virtual function which sends the message

Parameters

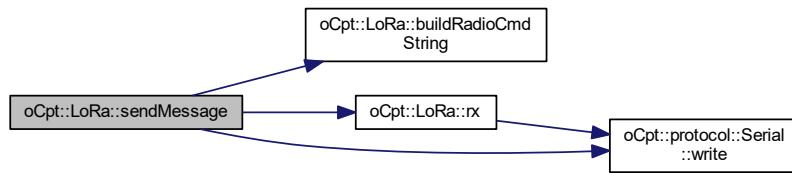
<i>msg</i>	the Message, consisting of a payload and a time stamp
------------	---

Implements [oCpt::iComm](#).

Definition at line 222 of file Communication.cpp.

References buildRadioCmdString(), NONE, PAUSE, oCpt::iComm::Message::Payload, rTX, rx(), sendAllowed_ ← , serial_, oCpt::iComm::Message::Stamp, oCpt::iComm::world_, and oCpt::protocol::Serial::write().

Here is the call graph for this function:



6.29.4.17 stop()

```
void oCpt::LoRa::stop ( ) [override], [virtual]
```

A pure virtual function which stops the communication device

Implements [oCpt::iComm](#).

Definition at line 185 of file Communication.cpp.

6.29.4.18 stringToHex()

```
void oCpt::LoRa::stringToHex (
    const std::string str,
    std::string & hexStr,
    const bool capital = true ) [protected]
```

Definition at line 245 of file Communication.cpp.

References boost::units::constants::c.

6.29.4.19 write()

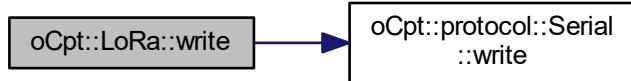
```
void oCpt::LoRa::write (
    const std::string & value ) [protected]
```

Definition at line 262 of file Communication.cpp.

References proceed_, serial_, and oCpt::protocol::Serial::write().

Referenced by initialize().

Here is the call graph for this function:



Here is the caller graph for this function:



6.29.5 Member Data Documentation

6.29.5.1 afcbw_

`BandWidth` `oCpt::LoRa::afcbw_` [protected]

Automatic frequency correction in kHz.

Definition at line 470 of file Communication.h.

Referenced by `initialize()`.

6.29.5.2 baudrate_

`unsigned int` `oCpt::LoRa::baudrate_` [protected]

Definition at line 465 of file Communication.h.

6.29.5.3 bw_

`RadioBandWidth` `oCpt::LoRa::bw_` [protected]

RadioBandWidth in kHz.

Definition at line 479 of file Communication.h.

Referenced by `calculateDownTime()`, and `initialize()`.

6.29.5.4 cr_

`CodingRate` `oCpt::LoRa::cr_` [protected]

The coding rate.

Definition at line 476 of file Communication.h.

Referenced by calculateDownTime(), and initialize().

6.29.5.5 crc_

`bool` `oCpt::LoRa::crc_` [protected]

CRC Header on or off.

Definition at line 475 of file Communication.h.

Referenced by calculateDownTime(), and initialize().

6.29.5.6 fdev_

`uint` `oCpt::LoRa::fdev_` [protected]

Frequency deviation between 0...200000.

Definition at line 473 of file Communication.h.

Referenced by initialize().

6.29.5.7 freq_

`unsigned long` `oCpt::LoRa::freq_` [protected]

Frequency between 433050000..4347900000 or 863000000..870000000.

Definition at line 467 of file Communication.h.

Referenced by initialize().

6.29.5.8 fskBitRate_

`uint` `oCpt::LoRa::fskBitRate_` [protected]

FSK bitrate between 1...300000.

Definition at line 472 of file Communication.h.

6.29.5.9 ignoreWarn_

```
bool oCpt::LoRa::ignoreWarn_ [protected]
```

Definition at line 462 of file Communication.h.

Referenced by initialize(), and messageReceived().

6.29.5.10 listen_

```
bool oCpt::LoRa::listen_ [protected]
```

Definition at line 463 of file Communication.h.

Referenced by initialize(), and messageReceived().

6.29.5.11 mod_

```
ModulationMode oCpt::LoRa::mod_ [protected]
```

Modulation mode.

Definition at line 466 of file Communication.h.

Referenced by initialize().

6.29.5.12 prlen_

```
uint oCpt::LoRa::prlen_ [protected]
```

Preamble length between 0...65535.

Definition at line 474 of file Communication.h.

Referenced by calculateDownTime(), and initialize().

6.29.5.13 proceed_

```
bool oCpt::LoRa::proceed_ [protected]
```

Definition at line 461 of file Communication.h.

Referenced by messageReceived(), and write().

6.29.5.14 pwr_

```
int8_t oCpt::LoRa::pwr_ [protected]
```

Power of transmission between -3...15.

Definition at line 468 of file Communication.h.

Referenced by initialize().

6.29.5.15 rxbw_

`BandWidth` `oCpt::LoRa::rbw_` [protected]

Signal bandwidth in kHz.

Definition at line 471 of file Communication.h.

Referenced by initialize().

6.29.5.16 sendAllowed_

`bool` `oCpt::LoRa::sendAllowed_` [protected]

Definition at line 480 of file Communication.h.

Referenced by sendMessage().

6.29.5.17 serial_

`protocol::Serial` `oCpt::LoRa::serial_` [protected]

SerialPort for UART communication with the chip.

Definition at line 464 of file Communication.h.

Referenced by initialize(), macpause(), messageReceived(), rx(), sendMessage(), and write().

6.29.5.18 sf_

`SpreadingFactor` `oCpt::LoRa::sf_` [protected]

Spreading factor of the signal.

Definition at line 469 of file Communication.h.

Referenced by calculateDownTime(), and initialize().

6.29.5.19 sync_

`unsigned int` `oCpt::LoRa::sync_` [protected]

Sync word.

Definition at line 478 of file Communication.h.

Referenced by initialize().

6.29.5.20 wdt_

```
unsigned long oCpt::LoRa::wdt_ [protected]
```

WatchDog 0...4294967295. Set to 0 to disable.

Definition at line 477 of file Communication.h.

Referenced by initialize().

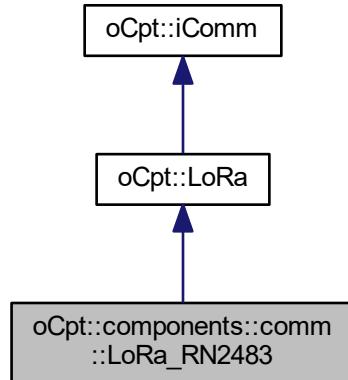
The documentation for this class was generated from the following files:

- /projects/mti/ohCaptain/ohCaptain/include/Core/[Communication.h](#)
- /projects/mti/ohCaptain/ohCaptain/src/Core/[Communication.cpp](#)

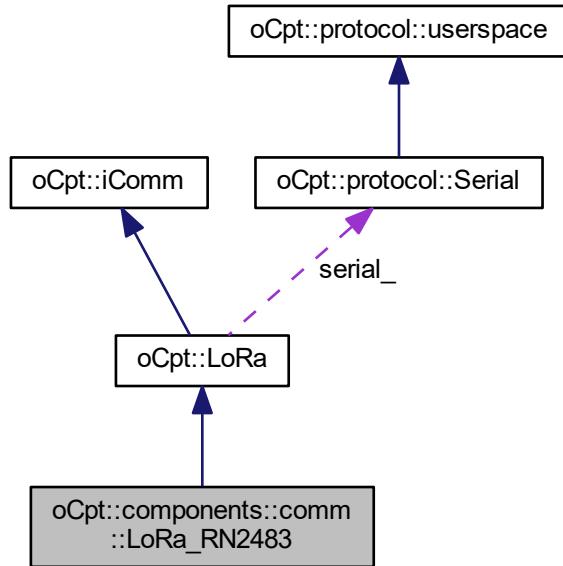
6.30 oCpt::components::comm::LoRa_RN2483 Class Reference

```
#include <LoRa_RN2483.h>
```

Inheritance diagram for oCpt::components::comm::LoRa_RN2483:



Collaboration diagram for oCpt::components::comm::LoRa_RN2483:



Public Member Functions

- `LoRa_RN2483 (const std::string &id, const std::string &device, World::ptr world=World::ptr(new World()), iController::io_t ioservice=iController::io_t(new boost::asio::io_service()))`
- `virtual ~LoRa_RN2483 ()`

Additional Inherited Members

6.30.1 Detailed Description

Definition at line 35 of file LoRa_RN2483.h.

6.30.2 Constructor & Destructor Documentation

6.30.2.1 LoRa_RN2483()

```

oCpt::components::comm::LoRa_RN2483::LoRa_RN2483 (
    const std::string & id,
    const std::string & device,
    World::ptr world = World::ptr(new World()),
    iController::io_t ioservice = iController::io_t(new boost::asio::io_service()) )

```

Definition at line 12 of file LoRa_RN2483.cpp.

6.30.2.2 ~LoRa_RN2483()

```
oCpt::components::comm::LoRa_RN2483::~LoRa_RN2483 ( ) [virtual]
```

Definition at line 18 of file LoRa_RN2483.cpp.

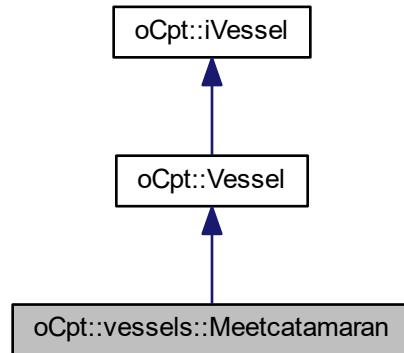
The documentation for this class was generated from the following files:

- /projects/mti/ohCaptain/ohCaptain/include/Communication/[LoRa_RN2483.h](#)
- /projects/mti/ohCaptain/ohCaptain/src/Communication/[LoRa_RN2483.cpp](#)

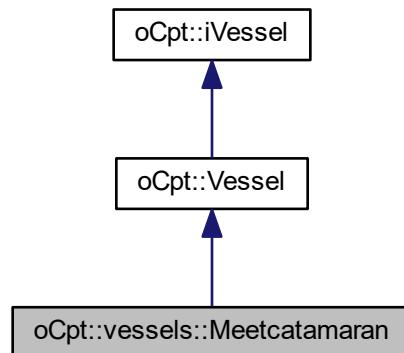
6.31 oCpt::vessels::Meetcatamaran Class Reference

```
#include <Meetcatamaran.h>
```

Inheritance diagram for oCpt::vessels::Meetcatamaran:



Collaboration diagram for oCpt::vessels::Meetcatamaran:



Public Member Functions

- [Meetcatamaran \(\)](#)
- virtual [~Meetcatamaran \(\)](#)

Additional Inherited Members

6.31.1 Detailed Description

Definition at line 11 of file Meetcatamaran.h.

6.31.2 Constructor & Destructor Documentation

6.31.2.1 [Meetcatamaran\(\)](#)

```
oCpt::vessels::Meetcatamaran::Meetcatamaran ( )
```

Definition at line 17 of file Meetcatamaran.cpp.

6.31.2.2 [~Meetcatamaran\(\)](#)

```
oCpt::vessels::Meetcatamaran::~Meetcatamaran ( ) [virtual]
```

Definition at line 31 of file Meetcatamaran.cpp.

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

- /projects/mti/ohCaptain/ohCaptain/include/Vessels/[Meetcatamaran.h](#)
- /projects/mti/ohCaptain/ohCaptain/src/Vessels/[Meetcatamaran.cpp](#)

6.32 oCpt::iComm::Message Struct Reference

```
#include <Communication.h>
```

Public Types

- [typedef boost::shared_ptr< Message > ptr](#)

Public Member Functions

- [Message \(const std::string &payload, const World::Time::timepoint_t &stamp\)](#)
- [~Message \(\)](#)

Public Attributes

- std::string [Payload](#)
- [World::Time::timepoint_t](#) [Stamp](#)

6.32.1 Detailed Description

A [Message](#) struct. Each device should receive and send messages with this type, consisting of Payload in the format of a string and a time when it was send or received

Definition at line 34 of file Communication.h.

6.32.2 Member Typedef Documentation

6.32.2.1 ptr

```
typedef boost::shared_ptr<Message> oCpt::iComm::Message::ptr
```

Definition at line 35 of file Communication.h.

6.32.3 Constructor & Destructor Documentation

6.32.3.1 [Message\(\)](#)

```
oCpt::iComm::Message::Message (
    const std::string & payload,
    const World::Time::timepoint\_t & stamp ) [inline]
```

A [Message](#) constructor taking the payload and the time

Parameters

<i>payload</i>	A string containing the payload TODO make generic with template
<i>stamp</i>	A time stamp, when the message was received, or is send

Definition at line 42 of file Communication.h.

6.32.3.2 [~Message\(\)](#)

```
oCpt::iComm::Message::~Message ( ) [inline]
```

The deconstructor

Definition at line 49 of file Communication.h.

References Payload.

6.32.4 Member Data Documentation

6.32.4.1 Payload

```
std::string oCpt::iComm::Message::Payload
```

Definition at line 49 of file Communication.h.

Referenced by oCpt::LoRa::sendMessage(), and ~Message().

6.32.4.2 Stamp

```
World::Time::timepoint_t oCpt::iComm::Message::Stamp
```

Definition at line 51 of file Communication.h.

Referenced by oCpt::LoRa::sendMessage().

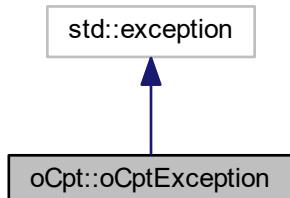
The documentation for this struct was generated from the following file:

- /projects/mti/ohCaptain/ohCaptain/include/Core/Communication.h

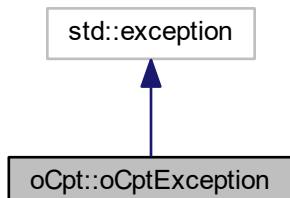
6.33 oCpt::oCptException Class Reference

```
#include <Exception.h>
```

Inheritance diagram for oCpt::oCptException:



Collaboration diagram for oCpt::oCptException:



Public Member Functions

- `oCptException` (`std::string msg="exception!", int id=-1`)
- `~oCptException () throw ()`
- `const char * what () const throw ()`

Private Attributes

- `std::string _msg`
- `int _id`

6.33.1 Detailed Description

Definition at line 13 of file Exception.h.

6.33.2 Constructor & Destructor Documentation

6.33.2.1 oCptException()

```
oCpt::oCptException::oCptException (
    std::string msg = "exception!",
    int id = -1 ) [inline]
```

Definition at line 15 of file Exception.h.

6.33.2.2 ~oCptException()

```
oCpt::oCptException::~oCptException ( ) throw () [inline]
```

Definition at line 17 of file Exception.h.

6.33.3 Member Function Documentation

6.33.3.1 what()

```
const char* oCpt::oCptException::what ( ) const throw () [inline]
```

Definition at line 19 of file Exception.h.

References `_msg`.

6.33.4 Member Data Documentation

6.33.4.1 _id

```
int oCpt::oCptException::_id [private]
```

Definition at line 23 of file Exception.h.

6.33.4.2 _msg

```
std::string oCpt::oCptException::_msg [private]
```

Definition at line 22 of file Exception.h.

Referenced by what().

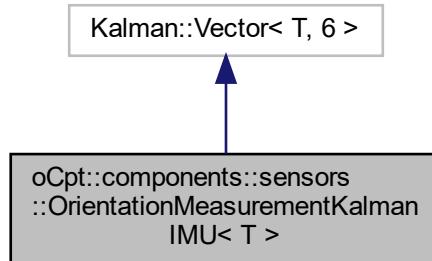
The documentation for this class was generated from the following file:

- /projects/mti/ohCaptain/ohCaptain/include/Core/[Exception.h](#)

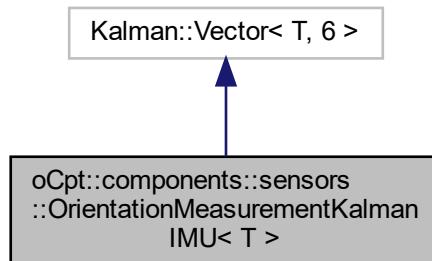
6.34 oCpt::components::sensors::OrientationMeasurementKalmanIMU< T > Class Template Reference

```
#include <KalmanIMU.h>
```

Inheritance diagram for oCpt::components::sensors::OrientationMeasurementKalmanIMU< T >:



Collaboration diagram for oCpt::components::sensors::OrientationMeasurementKalmanIMU< T >:



Public Types

- `typedef Kalman::Vector< T, 6 > Base`

Public Member Functions

- `OrientationMeasurementKalmanIMU (void)`
- `template<typename OtherDerived > OrientationMeasurementKalmanIMU (const Eigen::MatrixBase< OtherDerived > &other)`
- `template<typename OtherDerived > OrientationMeasurementKalmanIMU & operator= (const Eigen::MatrixBase< OtherDerived > &other)`
- `T theta () const`
- `T psi () const`
- `T phi () const`
- `T thetaPrime () const`
- `T psiPrime () const`
- `T phiPrime () const`
- `T & theta ()`
- `T & psi ()`
- `T & phi ()`
- `T & thetaPrime ()`
- `T & psiPrime ()`
- `T & phiPrime ()`

Static Public Attributes

- `static constexpr size_t Theta = 0`
- `static constexpr size_t Psi = 1`
- `static constexpr size_t Phi = 2`
- `static constexpr size_t ThetaPrime = 3`
- `static constexpr size_t PsiPrime = 4`
- `static constexpr size_t PhiPrime = 5`

6.34.1 Detailed Description

```
template<typename T>
class oCpt::components::sensors::OrientationMeasurementKalmanIMU< T >
```

Definition at line 120 of file KalmanIMU.h.

6.34.2 Member Typedef Documentation

6.34.2.1 Base

```
template<typename T >
typedef Kalman::Vector< T , 6 > oCpt::components::sensors::OrientationMeasurementKalmanIMU< T
>::Base
```

Definition at line 122 of file KalmanIMU.h.

6.34.3 Constructor & Destructor Documentation

6.34.3.1 OrientationMeasurementKalmanIMU() [1/2]

```
template<typename T >
oCpt::components::sensors::OrientationMeasurementKalmanIMU< T >::OrientationMeasurement<-
KalmanIMU (
    void ) [inline]
```

Definition at line 122 of file KalmanIMU.h.

6.34.3.2 OrientationMeasurementKalmanIMU() [2/2]

```
template<typename T >
template<typename OtherDerived >
oCpt::components::sensors::OrientationMeasurementKalmanIMU< T >::OrientationMeasurement<-
KalmanIMU (
    const Eigen::MatrixBase< OtherDerived > & other ) [inline]
```

Definition at line 122 of file KalmanIMU.h.

6.34.4 Member Function Documentation

6.34.4.1 operator=()

```
template<typename T >
template<typename OtherDerived >
OrientationMeasurementKalmanIMU& oCpt::components::sensors::OrientationMeasurementKalmanIMU< T
>::operator= (
    const Eigen::MatrixBase< OtherDerived > & other ) [inline]
```

Definition at line 122 of file KalmanIMU.h.

6.34.4.2 phi() [1/2]

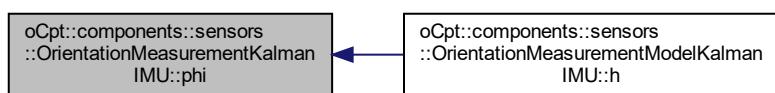
```
template<typename T >
T oCpt::components::sensors::OrientationMeasurementKalmanIMU< T >::phi ( ) const [inline]
```

Definition at line 135 of file KalmanIMU.h.

References oCpt::components::sensors::StateKalmanIMU< T >::Phi.

Referenced by oCpt::components::sensors::OrientationMeasurementModelKalmanIMU< T, CovarianceBase >::h().

Here is the caller graph for this function:



6.34.4.3 phi() [2/2]

```
template<typename T >
T& oCpt::components::sensors::OrientationMeasurementKalmanIMU< T >::phi ( ) [inline]
```

Definition at line 147 of file KalmanIMU.h.

References oCpt::components::sensors::StateKalmanIMU< T >::Phi.

6.34.4.4 phiPrime() [1/2]

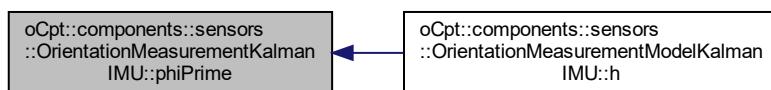
```
template<typename T >
T oCpt::components::sensors::OrientationMeasurementKalmanIMU< T >::phiPrime ( ) const [inline]
```

Definition at line 141 of file KalmanIMU.h.

References oCpt::components::sensors::StateKalmanIMU< T >::PhiPrime.

Referenced by oCpt::components::sensors::OrientationMeasurementModelKalmanIMU< T, CovarianceBase >↔
::h().

Here is the caller graph for this function:



6.34.4.5 phiPrime() [2/2]

```
template<typename T >
T& oCpt::components::sensors::OrientationMeasurementKalmanIMU< T >::phiPrime ( ) [inline]
```

Definition at line 153 of file KalmanIMU.h.

References oCpt::components::sensors::StateKalmanIMU< T >::PhiPrime.

6.34.4.6 psi() [1/2]

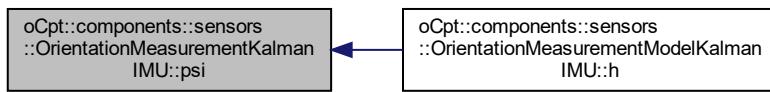
```
template<typename T >
T oCpt::components::sensors::OrientationMeasurementKalmanIMU< T >::psi ( ) const [inline]
```

Definition at line 133 of file KalmanIMU.h.

References oCpt::components::sensors::StateKalmanIMU< T >::Psi.

Referenced by oCpt::components::sensors::OrientationMeasurementModelKalmanIMU< T, CovarianceBase >↔::h().

Here is the caller graph for this function:



6.34.4.7 psi() [2/2]

```
template<typename T >
T& oCpt::components::sensors::OrientationMeasurementKalmanIMU< T >::psi ( ) [inline]
```

Definition at line 145 of file KalmanIMU.h.

References oCpt::components::sensors::StateKalmanIMU< T >::Psi.

6.34.4.8 psiPrime() [1/2]

```
template<typename T >
T oCpt::components::sensors::OrientationMeasurementKalmanIMU< T >::psiPrime ( ) const [inline]
```

Definition at line 139 of file KalmanIMU.h.

References oCpt::components::sensors::StateKalmanIMU< T >::PsiPrime.

Referenced by oCpt::components::sensors::OrientationMeasurementModelKalmanIMU< T, CovarianceBase >↔::h().

Here is the caller graph for this function:



6.34.4.9 psiPrime() [2/2]

```
template<typename T >
T& oCpt::components::sensors::OrientationMeasurementKalmanIMU< T >::psiPrime ( ) [inline]
```

Definition at line 151 of file KalmanIMU.h.

References oCpt::components::sensors::StateKalmanIMU< T >::PsiPrime.

6.34.4.10 theta() [1/2]

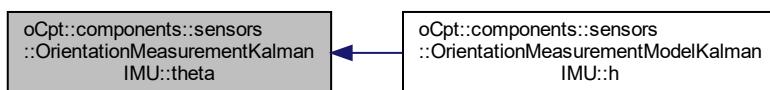
```
template<typename T >
T oCpt::components::sensors::OrientationMeasurementKalmanIMU< T >::theta ( ) const [inline]
```

Definition at line 131 of file KalmanIMU.h.

References oCpt::components::sensors::StateKalmanIMU< T >::Theta.

Referenced by oCpt::components::sensors::OrientationMeasurementModelKalmanIMU< T, CovarianceBase >↔::h().

Here is the caller graph for this function:



6.34.4.11 theta() [2/2]

```
template<typename T >
T& oCpt::components::sensors::OrientationMeasurementKalmanIMU< T >::theta ( ) [inline]
```

Definition at line 143 of file KalmanIMU.h.

References oCpt::components::sensors::StateKalmanIMU< T >::Theta.

6.34.4.12 thetaPrime() [1/2]

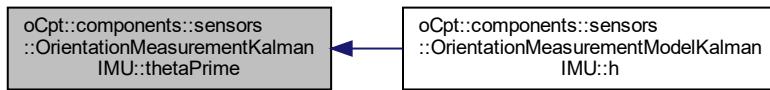
```
template<typename T >
T oCpt::components::sensors::OrientationMeasurementKalmanIMU< T >::thetaPrime () const [inline]
```

Definition at line 137 of file KalmanIMU.h.

References oCpt::components::sensors::StateKalmanIMU< T >::ThetaPrime.

Referenced by oCpt::components::sensors::OrientationMeasurementModelKalmanIMU< T, CovarianceBase >::h().

Here is the caller graph for this function:



6.34.4.13 thetaPrime() [2/2]

```
template<typename T >
T& oCpt::components::sensors::OrientationMeasurementKalmanIMU< T >::thetaPrime () [inline]
```

Definition at line 149 of file KalmanIMU.h.

References oCpt::components::sensors::StateKalmanIMU< T >::ThetaPrime.

6.34.5 Member Data Documentation

6.34.5.1 Phi

```
template<typename T >
constexpr size_t oCpt::components::sensors::OrientationMeasurementKalmanIMU< T >::Phi = 2
[static]
```

Definition at line 126 of file KalmanIMU.h.

6.34.5.2 PhiPrime

```
template<typename T >
constexpr size_t oCpt::components::sensors::OrientationMeasurementKalmanIMU< T >::PhiPrime = 5
[static]
```

Definition at line 129 of file KalmanIMU.h.

6.34.5.3 Psi

```
template<typename T >
constexpr size_t oCpt::components::sensors::OrientationMeasurementKalmanIMU< T >::Psi = 1
[static]
```

Definition at line 125 of file KalmanIMU.h.

6.34.5.4 PsiPrime

```
template<typename T >
constexpr size_t oCpt::components::sensors::OrientationMeasurementKalmanIMU< T >::PsiPrime = 4
[static]
```

Definition at line 128 of file KalmanIMU.h.

6.34.5.5 Theta

```
template<typename T >
constexpr size_t oCpt::components::sensors::OrientationMeasurementKalmanIMU< T >::Theta = 0
[static]
```

Definition at line 124 of file KalmanIMU.h.

6.34.5.6 ThetaPrime

```
template<typename T >
constexpr size_t oCpt::components::sensors::OrientationMeasurementKalmanIMU< T >::ThetaPrime =
3 [static]
```

Definition at line 127 of file KalmanIMU.h.

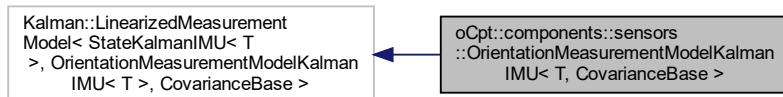
The documentation for this class was generated from the following file:

- /projects/mti/ohCaptain/ohCaptain/include/Sensors/[KalmanIMU.h](#)

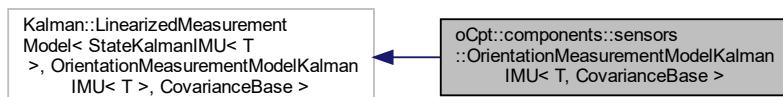
6.35 oCpt::components::sensors::OrientationMeasurementModelKalmanIMU< T, CovarianceBase > Class Template Reference

```
#include <KalmanIMU.h>
```

Inheritance diagram for oCpt::components::sensors::OrientationMeasurementModelKalmanIMU< T, CovarianceBase >:



Collaboration diagram for oCpt::components::sensors::OrientationMeasurementModelKalmanIMU< T, CovarianceBase >:



Public Types

- [typedef StateKalmanIMU< T > S](#)
- [typedef OrientationMeasurementKalmanIMU< T > M](#)

Public Member Functions

- [OrientationMeasurementModelKalmanIMU \(\)](#)
- [M h \(const S &x\) const](#)

6.35.1 Detailed Description

```
template<typename T, template< class > class CovarianceBase = Kalman::StandardBase>
class oCpt::components::sensors::OrientationMeasurementModelKalmanIMU< T, CovarianceBase >
```

Definition at line 157 of file KalmanIMU.h.

6.35.2 Member Typedef Documentation

6.35.2.1 M

```
template<typename T , template< class > class CovarianceBase = Kalman::StandardBase>
typedef OrientationMeasurementKalmanIMU<T> oCpt::components::sensors::OrientationMeasurement<→
ModelKalmanIMU< T, CovarianceBase >::M
```

Definition at line 162 of file KalmanIMU.h.

6.35.2.2 S

```
template<typename T , template< class > class CovarianceBase = Kalman::StandardBase>
typedef StateKalmanIMU<T> oCpt::components::sensors::OrientationMeasurementModelKalmanIMU< T,
CovarianceBase >::S
```

Definition at line 160 of file KalmanIMU.h.

6.35.3 Constructor & Destructor Documentation

6.35.3.1 OrientationMeasurementModelKalmanIMU()

```
template<typename T , template< class > class CovarianceBase = Kalman::StandardBase>
oCpt::components::sensors::OrientationMeasurementModelKalmanIMU< T, CovarianceBase >::Orientation<→
MeasurementModelKalmanIMU ( ) [inline]
```

Definition at line 164 of file KalmanIMU.h.

6.35.4 Member Function Documentation

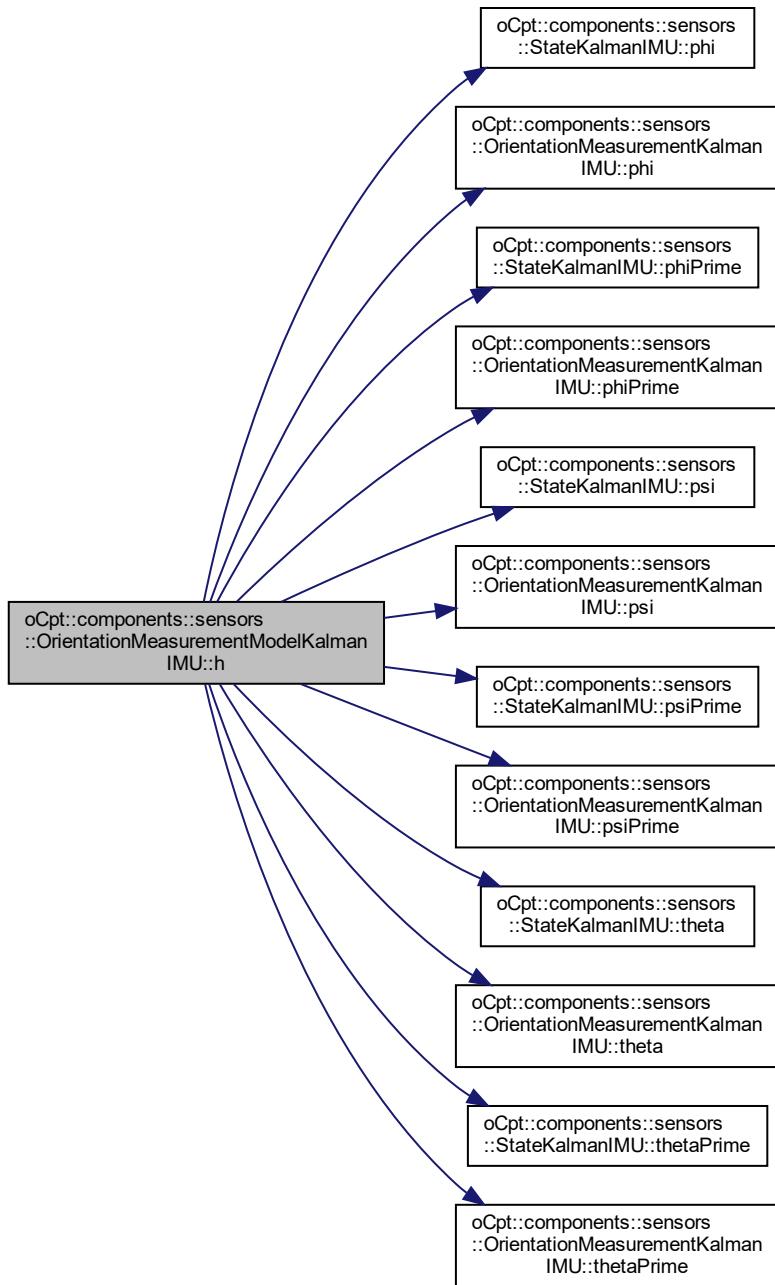
6.35.4.1 h()

```
template<typename T , template< class > class CovarianceBase = Kalman::StandardBase>
M oCpt::components::sensors::OrientationMeasurementModelKalmanIMU< T, CovarianceBase >::h (
    const S & x ) const [inline]
```

Definition at line 169 of file KalmanIMU.h.

References oCpt::components::sensors::StateKalmanIMU< T >::phi(), oCpt::components::sensors::Orientation<→
MeasurementKalmanIMU< T >::phi(), oCpt::components::sensors::StateKalmanIMU< T >::phiPrime(), oCpt<→
::components::sensors::OrientationMeasurementKalmanIMU< T >::phiPrime(), oCpt::components::sensors<→
::StateKalmanIMU< T >::psi(), oCpt::components::sensors::OrientationMeasurementKalmanIMU< T >::psi(), oCpt::components::sensors::StateKalmanIMU< T >::psiPrime(), oCpt::components::sensors::Orientation<→
MeasurementKalmanIMU< T >::psiPrime(), oCpt::components::sensors::StateKalmanIMU< T >::theta(), oCpt<→
::components::sensors::OrientationMeasurementKalmanIMU< T >::theta(), oCpt::components::sensors<→
::StateKalmanIMU< T >::thetaPrime(), and oCpt::components::sensors::OrientationMeasurementKalmanIMU< T >::thetaPrime().

Here is the call graph for this function:



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

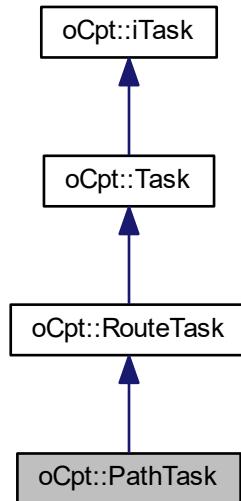
- /projects/mti/ohCaptain/ohCaptain/include/Sensors/[KalmanIMU.h](#)

6.36 oCpt::PathTask Class Reference

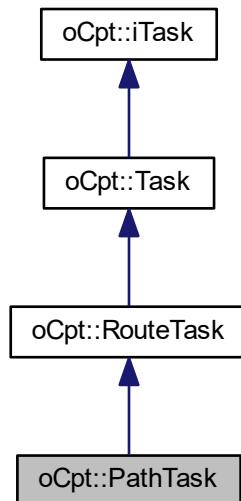
An object representing a normal A to B type of path planning.

```
#include <Task.h>
```

Inheritance diagram for oCpt::PathTask:



Collaboration diagram for oCpt::PathTask:



Public Member Functions

- `PathTask (Vessel::ptr vessel, bool concurrent=false)`
- `virtual ~PathTask ()`

Additional Inherited Members

6.36.1 Detailed Description

An object representing a normal A to B type of path planning.

All these types of tasks need to plan an optimum route between A and B, either in time, energy consumption or

Definition at line 244 of file Task.h.

6.36.2 Constructor & Destructor Documentation

6.36.2.1 PathTask()

```
oCpt::PathTask::PathTask (
    Vessel::ptr vessel,
    bool concurrent = false )
```

Constructor of the interface

Returns

Definition at line 61 of file Task.cpp.

6.36.2.2 ~PathTask()

```
oCpt::PathTask::~PathTask ( ) [virtual]
```

The deconstructor

Definition at line 63 of file Task.cpp.

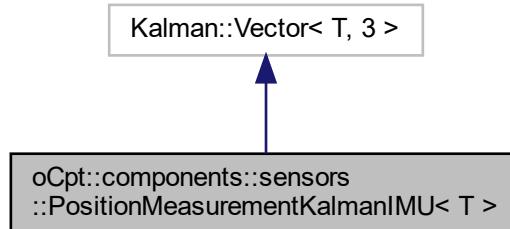
The documentation for this class was generated from the following files:

- /projects/mti/ohCaptain/ohCaptain/include/Core/[Task.h](#)
- /projects/mti/ohCaptain/ohCaptain/src/Core/[Task.cpp](#)

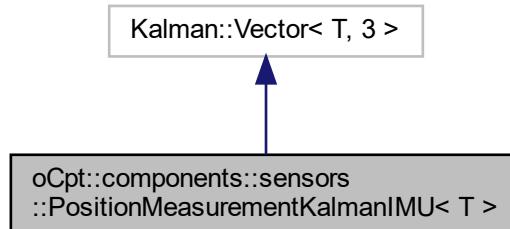
6.37 oCpt::components::sensors::PositionMeasurementKalmanIMU< T > Class Template Reference

```
#include <KalmanIMU.h>
```

Inheritance diagram for oCpt::components::sensors::PositionMeasurementKalmanIMU< T >:



Collaboration diagram for oCpt::components::sensors::PositionMeasurementKalmanIMU< T >:



Public Types

- `typedef Kalman::Vector< T, 3 > Base`

Public Member Functions

- `PositionMeasurementKalmanIMU (void)`
- `template<typename OtherDerived > PositionMeasurementKalmanIMU (const Eigen::MatrixBase< OtherDerived > &other)`
- `template<typename OtherDerived > PositionMeasurementKalmanIMU & operator= (const Eigen::MatrixBase< OtherDerived > &other)`
- `T acc_x () const`
- `T acc_y () const`
- `T acc_z () const`
- `T & acc_x ()`
- `T & acc_y ()`
- `T & acc_z ()`

Static Public Attributes

- static constexpr size_t `accX` = 0
- static constexpr size_t `accY` = 1
- static constexpr size_t `accZ` = 2

6.37.1 Detailed Description

```
template<typename T>
class oCpt::components::sensors::PositionMeasurementKalmanIMU< T >
```

Definition at line 185 of file KalmanIMU.h.

6.37.2 Member Typedef Documentation

6.37.2.1 Base

```
template<typename T>
typedef Kalman::Vector< T , 3 > oCpt::components::sensors::PositionMeasurementKalmanIMU< T
>::Base
```

Definition at line 187 of file KalmanIMU.h.

6.37.3 Constructor & Destructor Documentation

6.37.3.1 PositionMeasurementKalmanIMU() [1/2]

```
template<typename T>
oCpt::components::sensors::PositionMeasurementKalmanIMU< T >::PositionMeasurementKalmanIMU (
    void ) [inline]
```

Definition at line 187 of file KalmanIMU.h.

6.37.3.2 PositionMeasurementKalmanIMU() [2/2]

```
template<typename T>
template<typename OtherDerived >
oCpt::components::sensors::PositionMeasurementKalmanIMU< T >::PositionMeasurementKalmanIMU (
    const Eigen::MatrixBase< OtherDerived > & other ) [inline]
```

Definition at line 187 of file KalmanIMU.h.

6.37.4 Member Function Documentation

6.37.4.1 acc_x() [1/2]

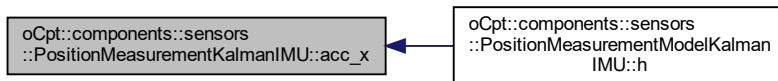
```
template<typename T>
T oCpt::components::sensors::PositionMeasurementKalmanIMU< T >::acc_x ( ) const [inline]
```

Definition at line 193 of file KalmanIMU.h.

References oCpt::components::sensors::StateKalmanIMU< T >::accX.

Referenced by oCpt::components::sensors::PositionMeasurementModelKalmanIMU< T, CovarianceBase >::h().

Here is the caller graph for this function:



6.37.4.2 acc_x() [2/2]

```
template<typename T>
T& oCpt::components::sensors::PositionMeasurementKalmanIMU< T >::acc_x ( ) [inline]
```

Definition at line 199 of file KalmanIMU.h.

References oCpt::components::sensors::StateKalmanIMU< T >::accX.

6.37.4.3 acc_y() [1/2]

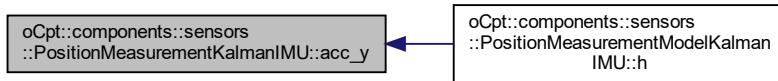
```
template<typename T>
T oCpt::components::sensors::PositionMeasurementKalmanIMU< T >::acc_y ( ) const [inline]
```

Definition at line 195 of file KalmanIMU.h.

References oCpt::components::sensors::StateKalmanIMU< T >::accY.

Referenced by oCpt::components::sensors::PositionMeasurementModelKalmanIMU< T, CovarianceBase >::h().

Here is the caller graph for this function:



6.37.4.4 acc_y() [2/2]

```
template<typename T>
T& oCpt::components::sensors::PositionMeasurementKalmanIMU< T >::acc_y ( ) [inline]
```

Definition at line 201 of file KalmanIMU.h.

References oCpt::components::sensors::StateKalmanIMU< T >::accY.

6.37.4.5 acc_z() [1/2]

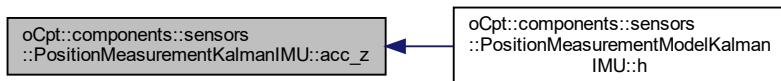
```
template<typename T>
T oCpt::components::sensors::PositionMeasurementKalmanIMU< T >::acc_z ( ) const [inline]
```

Definition at line 197 of file KalmanIMU.h.

References oCpt::components::sensors::StateKalmanIMU< T >::accZ.

Referenced by oCpt::components::sensors::PositionMeasurementModelKalmanIMU< T, CovarianceBase >::h().

Here is the caller graph for this function:



6.37.4.6 acc_z() [2/2]

```
template<typename T>
T& oCpt::components::sensors::PositionMeasurementKalmanIMU< T >::acc_z ( ) [inline]
```

Definition at line 203 of file KalmanIMU.h.

References oCpt::components::sensors::StateKalmanIMU< T >::accZ.

6.37.4.7 operator=()

```
template<typename T>
template<typename OtherDerived >
PositionMeasurementKalmanIMU& oCpt::components::sensors::PositionMeasurementKalmanIMU< T >::operator= (
    const Eigen::MatrixBase< OtherDerived > & other) [inline]
```

Definition at line 187 of file KalmanIMU.h.

6.37.5 Member Data Documentation

6.37.5.1 accX

```
template<typename T>
constexpr size_t oCpt::components::sensors::PositionMeasurementKalmanIMU< T >::accX = 0 [static]
```

Definition at line 189 of file KalmanIMU.h.

6.37.5.2 accY

```
template<typename T>
constexpr size_t oCpt::components::sensors::PositionMeasurementKalmanIMU< T >::accY = 1 [static]
```

Definition at line 190 of file KalmanIMU.h.

6.37.5.3 accZ

```
template<typename T>
constexpr size_t oCpt::components::sensors::PositionMeasurementKalmanIMU< T >::accZ = 2 [static]
```

Definition at line 191 of file KalmanIMU.h.

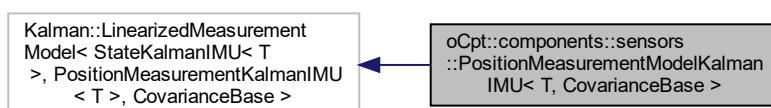
The documentation for this class was generated from the following file:

- /projects/mti/ohCaptain/ohCaptain/include/Sensors/KalmanIMU.h

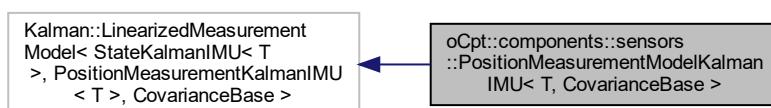
6.38 oCpt::components::sensors::PositionMeasurementModelKalmanIMU< T, CovarianceBase > Class Template Reference

```
#include <KalmanIMU.h>
```

Inheritance diagram for oCpt::components::sensors::PositionMeasurementModelKalmanIMU< T, CovarianceBase >:



Collaboration diagram for oCpt::components::sensors::PositionMeasurementModelKalmanIMU< T, CovarianceBase >:



Public Types

- `typedef StateKalmanIMU< T > S`
- `typedef PositionMeasurementKalmanIMU< T > M`

Public Member Functions

- `PositionMeasurementModelKalmanIMU ()`
- `M h (const S &x) const`

6.38.1 Detailed Description

```
template<typename T, template< class > class CovarianceBase = Kalman::StandardBase>
class oCpt::components::sensors::PositionMeasurementModelKalmanIMU< T, CovarianceBase >
```

Definition at line 207 of file KalmanIMU.h.

6.38.2 Member Typedef Documentation

6.38.2.1 M

```
template<typename T , template< class > class CovarianceBase = Kalman::StandardBase>
typedef PositionMeasurementKalmanIMU<T> oCpt::components::sensors::PositionMeasurementModel< KalmanIMU< T, CovarianceBase >>::M
```

Definition at line 212 of file KalmanIMU.h.

6.38.2.2 S

```
template<typename T , template< class > class CovarianceBase = Kalman::StandardBase>
typedef StateKalmanIMU<T> oCpt::components::sensors::PositionMeasurementModelKalmanIMU< T,
CovarianceBase >>::S
```

Definition at line 210 of file KalmanIMU.h.

6.38.3 Constructor & Destructor Documentation

6.38.3.1 PositionMeasurementModelKalmanIMU()

```
template<typename T , template< class > class CovarianceBase = Kalman::StandardBase>
oCpt::components::sensors::PositionMeasurementModelKalmanIMU< T, CovarianceBase >>::Position< MeasurementModelKalmanIMU ( ) [inline]
```

Definition at line 214 of file KalmanIMU.h.

6.38.4 Member Function Documentation

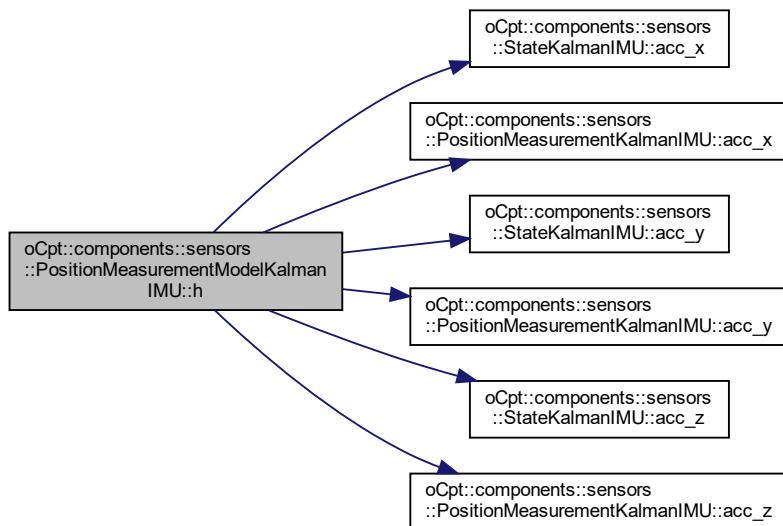
6.38.4.1 h()

```
template<typename T , template< class > class CovarianceBase = Kalman::StandardBase>
M oCpt::components::sensors::PositionMeasurementModelKalmanIMU< T, CovarianceBase >::h (
    const S & x ) const [inline]
```

Definition at line 218 of file KalmanIMU.h.

References oCpt::components::sensors::StateKalmanIMU< T >::acc_x(), oCpt::components::sensors::PositionMeasurementKalmanIMU< T >::acc_x(), oCpt::components::sensors::StateKalmanIMU< T >::acc_y(), oCpt::components::sensors::PositionMeasurementKalmanIMU< T >::acc_y(), oCpt::components::sensors::StateKalmanIMU< T >::acc_z(), and oCpt::components::sensors::PositionMeasurementKalmanIMU< T >::acc_z().

Here is the call graph for this function:



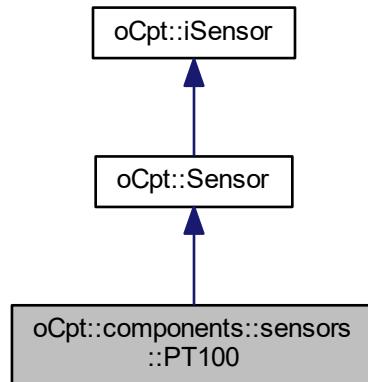
The documentation for this class was generated from the following file:

- /projects/mti/ohCaptain/ohCaptain/include/Sensors/[KalmanIMU.h](#)

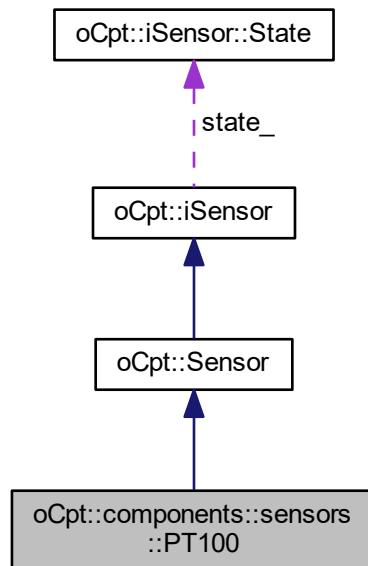
6.39 oCpt::components::sensors::PT100 Class Reference

```
#include <PT100.h>
```

Inheritance diagram for oCpt::components::sensors::PT100:



Collaboration diagram for oCpt::components::sensors::PT100:



Public Types

- `typedef quantity< celsius::temperature, double > ReturnValue_t`

Public Member Functions

- `PT100 (iController::ptr controller, World::ptr world, std::string id, uint8_t pinid, uint8_t device)`
- `~PT100 ()`
- `void updateSensor ()`
- `void run ()`
- `void stop ()`
- `void init ()`
- `void setCalibrationTemperature (std::pair< ReturnValue_t, ReturnValue_t > temperature, std::pair< uint16_t, uint16_t > analogeValue)`

Private Attributes

- `uint16_t _analogeValue`
- `uint8_t _device = 0`
- `uint8_t _pinid = 0`
- `ReturnValue_t _dy_dx = 1.0 * celsius::degree`
- `ReturnValue_t _constant = 0.0 * celsius::degree`

Additional Inherited Members

6.39.1 Detailed Description

A standard `PT100` sensors

Definition at line 19 of file PT100.h.

6.39.2 Member Typedef Documentation

6.39.2.1 ReturnValue_t

```
typedef quantity<celsius::temperature, double> oCpt::components::sensors::PT100::ReturnValue_t
```

Definition at line 21 of file PT100.h.

6.39.3 Constructor & Destructor Documentation

6.39.3.1 PT100()

```
oCpt::components::sensors::PT100::PT100 (
    iController::ptr controller,
    World::ptr world,
    std::string id,
    uint8_t pinid,
    uint8_t device )
```

Definition at line 12 of file PT100.cpp.

6.39.3.2 ~PT100()

```
oCpt::components::sensors::PT100::~PT100 ( )
```

Definition at line 19 of file PT100.cpp.

6.39.4 Member Function Documentation

6.39.4.1 init()

```
void oCpt::components::sensors::PT100::init ( ) [virtual]
```

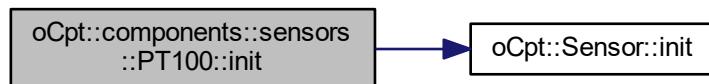
Initialize the sensor

Reimplemented from [oCpt::Sensor](#).

Definition at line 48 of file PT100.cpp.

References [oCpt::Sensor::init\(\)](#).

Here is the call graph for this function:



6.39.4.2 run()

```
void oCpt::components::sensors::PT100::run ( ) [virtual]
```

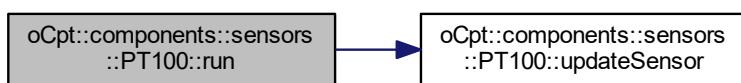
virtual function starting the run service for the IO

Reimplemented from [oCpt::Sensor](#).

Definition at line 39 of file PT100.cpp.

References [oCpt::ISensor::sig_](#), and [updateSensor\(\)](#).

Here is the call graph for this function:



6.39.4.3 setCalibrationTemperature()

```
void oCpt::components::sensors::PT100::setCalibrationTemperature (
    std::pair< ReturnValue_t, ReturnValue_t > temperature,
    std::pair< uint16_t, uint16_t > analogeValue )
```

Definition at line 31 of file PT100.cpp.

References `_constant`, and `_dy_dx`.

6.39.4.4 stop()

```
void oCpt::components::sensors::PT100::stop () [virtual]
```

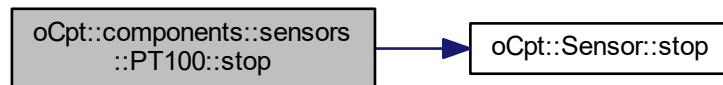
virtual function stopping the run

Reimplemented from [oCpt::Sensor](#).

Definition at line 44 of file PT100.cpp.

References [oCpt::Sensor::stop\(\)](#).

Here is the call graph for this function:



6.39.4.5 updateSensor()

```
void oCpt::components::sensors::PT100::updateSensor () [virtual]
```

virtual function which performs a sensor update, obtaining a new value and sending a signal afterwards

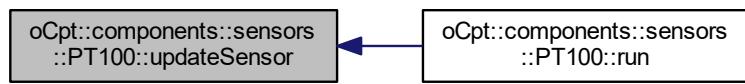
Reimplemented from [oCpt::Sensor](#).

Definition at line 23 of file PT100.cpp.

References `_analogeValue`, `_constant`, `_dy_dx`, `_pinid`, [oCpt::iSensor::controller_](#), [oCpt::iSensor::State::Stamp](#), [oCpt::iSensor::state_](#), [oCpt::iSensor::State::Value](#), and [oCpt::iSensor::world_](#).

Referenced by `run()`.

Here is the caller graph for this function:



6.39.5 Member Data Documentation

6.39.5.1 _analogeValue

```
uint16_t oCpt::components::sensors::PT100::_analogeValue [private]
```

Definition at line 39 of file PT100.h.

Referenced by updateSensor().

6.39.5.2 _constant

```
ReturnValue_t oCpt::components::sensors::PT100::_constant = 0.0 * celsius::degree [private]
```

Definition at line 43 of file PT100.h.

Referenced by setCalibrationTemperature(), and updateSensor().

6.39.5.3 _device

```
uint8_t oCpt::components::sensors::PT100::_device = 0 [private]
```

Definition at line 40 of file PT100.h.

6.39.5.4 _dy_dx

```
ReturnValue_t oCpt::components::sensors::PT100::_dy_dx = 1.0 * celsius::degree [private]
```

Definition at line 42 of file PT100.h.

Referenced by setCalibrationTemperature(), and updateSensor().

6.39.5.5 _pinid

```
uint8_t oCpt::components::sensors::PT100::_pinid = 0 [private]
```

Definition at line 41 of file PT100.h.

Referenced by updateSensor().

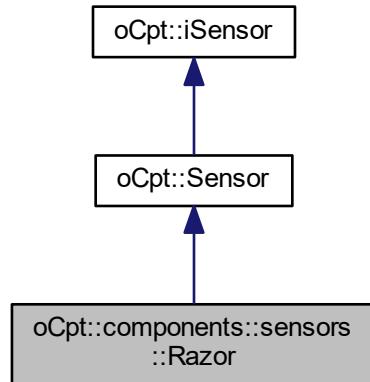
The documentation for this class was generated from the following files:

- /projects/mti/ohCaptain/ohCaptain/include/Sensors/[PT100.h](#)
- /projects/mti/ohCaptain/ohCaptain/src/Sensors/[PT100.cpp](#)

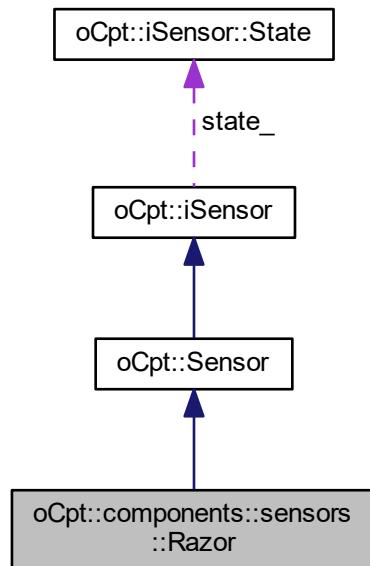
6.40 oCpt::components::sensors::Razor Class Reference

```
#include <Razor.h>
```

Inheritance diagram for oCpt::components::sensors::Razor:



Collaboration diagram for oCpt::components::sensors::Razor:



Classes

- struct [ReturnValue](#)

Public Types

- enum `Mode` { `CONT` = 0, `REQ` = 1 }
- typedef quantity< si::magnetic_flux_density, float > `Magnetic_flux_density_t`
- typedef quantity< si::angular_velocity, float > `Angular_velocity_t`
- typedef quantity< si::acceleration, float > `Acceleration_t`
- typedef struct `oCpt::components::sensors::Razor::ReturnValue` `ReturnValue_t`

Public Member Functions

- `Razor` (`iController::ptr` controller, `World::ptr` world, std::string id, std::string device, unsigned int baudrate, `Mode` mode=`Mode::CONT`, uint8_t freq=50)
- `~Razor` ()
- void `updateSensor` ()
- void `run` ()
- void `stop` ()
- void `init` ()
- void `setIoservice` (boost::shared_ptr< boost::asio::io_service > ioservice)
- `Mode` `getMode` () const
- void `setMode` (`Mode` mode)
- uint8_t `getFreq` () const
- void `setFreq` (uint8_t freq)

Private Member Functions

- void `fillReturnValue` (`ReturnValue_t` &retVal, float *values)
- void `msgHandler` (const unsigned char *data, size_t size)
- bool `checkLRC` (std::vector< char *> data)

Private Attributes

- std::string `device_`
- `protocol::Serial::ptr` `serial_`
- `protocol::Serial::cb_func` `cb`
- `Mode` `mode_`
- uint8_t `freq_` = 50

Additional Inherited Members

6.40.1 Detailed Description

Definition at line 15 of file Razor.h.

6.40.2 Member Typedef Documentation

6.40.2.1 Acceleration_t

```
typedef quantity<si::acceleration, float> oCpt::components::sensors::Razor::Acceleration_t
```

Definition at line 19 of file Razor.h.

6.40.2.2 Angular_velocity_t

```
typedef quantity<si::angular_velocity, float> oCpt::components::sensors::Razor::Angular←  
velocity_t
```

Definition at line 18 of file Razor.h.

6.40.2.3 Magnetic_flux_density_t

```
typedef quantity<si::magnetic_flux_density, float> oCpt::components::sensors::Razor::Magnetic←  
_flux_density_t
```

Definition at line 17 of file Razor.h.

6.40.2.4 ReturnValue_t

```
typedef struct oCpt::components::sensors::Razor::ReturnValue oCpt::components::sensors::←  
Razor::ReturnValue_t
```

6.40.3 Member Enumeration Documentation

6.40.3.1 Mode

```
enum oCpt::components::sensors::Razor::Mode
```

Enumerator

CONT	
REQ	

Definition at line 27 of file Razor.h.

6.40.4 Constructor & Destructor Documentation

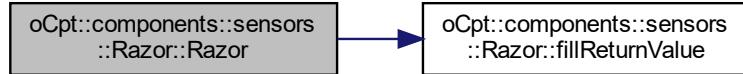
6.40.4.1 Razor()

```
oCpt::components::sensors::Razor::Razor (  
    iController::ptr controller,  
    World::ptr world,  
    std::string id,  
    std::string device,  
    unsigned int baudrate,  
    Mode mode = Mode::CONT,  
    uint8_t freq = 50 )
```

Definition at line 13 of file Razor.cpp.

References `fillReturnValue()`, `serial_`, `oCpt::iSensor::State::Stamp`, `oCpt::iSensor::state_`, `oCpt::iSensor::State::←Value`, and `oCpt::iSensor::world_`.

Here is the call graph for this function:



6.40.4.2 ~Razor()

`oCpt::components::sensors::Razor::~Razor()`

Definition at line 35 of file `Razor.cpp`.

References `serial_`.

6.40.5 Member Function Documentation

6.40.5.1 checkLRC()

```
bool oCpt::components::sensors::Razor::checkLRC (
    std::vector< char * > data ) [private]
```

Definition at line 120 of file `Razor.cpp`.

6.40.5.2 fillReturnValue()

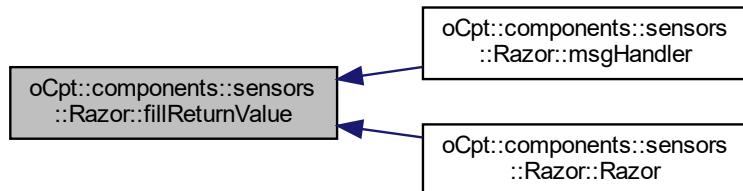
```
void oCpt::components::sensors::Razor::fillReturnValue (
    Razor::ReturnValue_t & retVal,
    float * values ) [private]
```

Definition at line 85 of file `Razor.cpp`.

References `oCpt::components::sensors::Razor::ReturnValue::acc`, `oCpt::components::sensors::Razor::ReturnValue::gyro`, and `oCpt::components::sensors::Razor::ReturnValue::mag`.

Referenced by `msgHandler()`, and `Razor()`.

Here is the caller graph for this function:



6.40.5.3 `getFreq()`

```
uint8_t oCpt::components::sensors::Razor::getFreq ( ) const
```

Definition at line 144 of file Razor.cpp.

References freq_.

6.40.5.4 `getMode()`

```
Razor::Mode oCpt::components::sensors::Razor::getMode ( ) const
```

Definition at line 131 of file Razor.cpp.

References mode_.

6.40.5.5 `init()`

```
void oCpt::components::sensors::Razor::init ( ) [virtual]
```

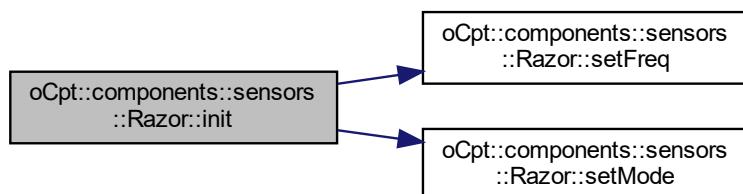
Initialize the sensor

Reimplemented from [oCpt::Sensor](#).

Definition at line 75 of file Razor.cpp.

References cb, freq_, mode_, serial_, setFreq(), and setMode().

Here is the call graph for this function:



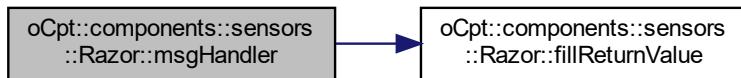
6.40.5.6 msgHandler()

```
void oCpt::components::sensors::Razor::msgHandler (
    const unsigned char * data,
    size_t size ) [private]
```

Definition at line 98 of file Razor.cpp.

References `fillReturnValue()`, `oCpt::iSensor::sig_`, `oCpt::iSensor::State::Stamp`, `oCpt::iSensor::state_`, `oCpt::iSensor::State::Value`, and `oCpt::iSensor::world_`.

Here is the call graph for this function:



6.40.5.7 run()

```
void oCpt::components::sensors::Razor::run () [virtual]
```

virtual function starting the run service for the IO

Reimplemented from [oCpt::Sensor](#).

Definition at line 53 of file Razor.cpp.

References `oCpt::Sensor::run()`, `oCpt::iSensor::sensorRunning_`, and `serial_`.

Here is the call graph for this function:



6.40.5.8 setFreq()

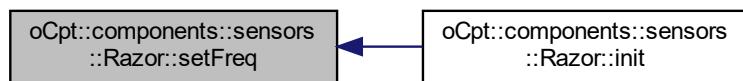
```
void oCpt::components::sensors::Razor::setFreq (
    uint8_t freq )
```

Definition at line 148 of file Razor.cpp.

References freq_, and serial_.

Referenced by init().

Here is the caller graph for this function:



6.40.5.9 setIoservice()

```
void oCpt::components::sensors::Razor::setIoservice (
    boost::shared_ptr< boost::asio::io_service > ioservice ) [virtual]
```

Setting the used Asynchronous Input Output service

Parameters

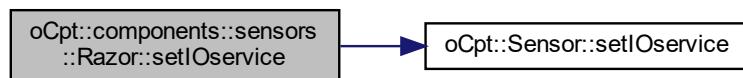
<i>ioservice</i>	ASIO IO service, which handles the async calls from multiple sensors
------------------	--

Reimplemented from [oCpt::Sensor](#).

Definition at line 70 of file Razor.cpp.

References serial_, and [oCpt::Sensor::setIoservice\(\)](#).

Here is the call graph for this function:



6.40.5.10 setMode()

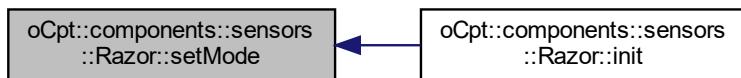
```
void oCpt::components::sensors::Razor::setMode (   
    Razor::Mode mode )
```

Definition at line 135 of file Razor.cpp.

References mode_, and serial_.

Referenced by init().

Here is the caller graph for this function:



6.40.5.11 stop()

```
void oCpt::components::sensors::Razor::stop ( ) [virtual]
```

virtual function stopping the run

Reimplemented from [oCpt::Sensor](#).

Definition at line 60 of file Razor.cpp.

References [oCpt::iSensor::sensorRunning_](#), [serial_](#), and [oCpt::Sensor::stop\(\)](#).

Here is the call graph for this function:



6.40.5.12 updateSensor()

```
void oCpt::components::sensors::Razor::updateSensor ( ) [virtual]
```

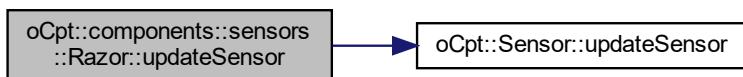
virtual function which performs a sensor update, obtaining a new value and sending a signal afterwards

Reimplemented from [oCpt::Sensor](#).

Definition at line 41 of file Razor.cpp.

References mode_, serial_, [oCpt::iSensor::State::Stamp](#), [oCpt::iSensor::state_](#), [oCpt::Sensor::updateSensor\(\)](#), and [oCpt::iSensor::world_](#)

Here is the call graph for this function:



6.40.6 Member Data Documentation

6.40.6.1 cb

```
protocol::Serial::cb_func oCpt::components::sensors::Razor::cb [private]
```

Definition at line 50 of file Razor.h.

Referenced by [init\(\)](#).

6.40.6.2 device_

```
std::string oCpt::components::sensors::Razor::device_ [private]
```

Definition at line 48 of file Razor.h.

6.40.6.3 freq_

```
uint8_t oCpt::components::sensors::Razor::freq_ = 50 [private]
```

Definition at line 62 of file Razor.h.

Referenced by [getFreq\(\)](#), [init\(\)](#), and [setFreq\(\)](#).

6.40.6.4 mode_

```
Mode oCpt::components::sensors::Razor::mode_ [private]
```

Definition at line 51 of file Razor.h.

Referenced by getMode(), init(), setMode(), and updateSensor().

6.40.6.5 serial_

```
protocol::Serial::ptr oCpt::components::sensors::Razor::serial_ [private]
```

Definition at line 49 of file Razor.h.

Referenced by init(), Razor(), run(), setFreq(), setIoservice(), setMode(), stop(), updateSensor(), and ~Razor().

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

- /projects/mti/ohCaptain/ohCaptain/include/Sensors/[Razor.h](#)
- /projects/mti/ohCaptain/ohCaptain/src/Sensors/[Razor.cpp](#)

6.41 oCpt::components::sensors::Razor::ReturnValue Struct Reference

```
#include <Razor.h>
```

Public Attributes

- [Angular_velocity_t gyro](#) [3]
- [Magnetic_flux_density_t mag](#) [3]
- [Acceleration_t acc](#) [3]

6.41.1 Detailed Description

Definition at line 21 of file Razor.h.

6.41.2 Member Data Documentation

6.41.2.1 acc

```
Acceleration_t oCpt::components::sensors::Razor::ReturnValue::acc [3]
```

Definition at line 24 of file Razor.h.

Referenced by oCpt::components::sensors::Razor::fillReturnValue(), and oCpt::components::sensors::KalmanIMU::RazorUpdate().

6.41.2.2 gyro

```
Angular_velocity_t oCpt::components::sensors::Razor::ReturnValue::gyro[3]
```

Definition at line 22 of file Razor.h.

Referenced by oCpt::components::sensors::Razor::fillReturnValue().

6.41.2.3 mag

```
Magnetic_flux_density_t oCpt::components::sensors::Razor::ReturnValue::mag[3]
```

Definition at line 23 of file Razor.h.

Referenced by oCpt::components::sensors::Razor::fillReturnValue().

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

- /projects/mti/ohCaptain/ohCaptain/include/Sensors/[Razor.h](#)

6.42 oCpt::components::sensors::KalmanIMU::ReturnValue Struct Reference

```
#include <KalmanIMU.h>
```

Public Attributes

- quantity< si::length, double > [position](#) [2]
- quantity< si::velocity, double > [velocity](#) [2]
- quantity< si::acceleration, double > [acceleration](#) [2]
- quantity< si::plane_angle, double > [orientation](#) [3]
- quantity< angular_velocity_dimension, double > [orientation_change](#) [3]

6.42.1 Detailed Description

Definition at line 231 of file KalmanIMU.h.

6.42.2 Member Data Documentation

6.42.2.1 acceleration

```
quantity<si::acceleration, double> oCpt::components::sensors::KalmanIMU::ReturnValue::acceleration[2]
```

Definition at line 234 of file KalmanIMU.h.

6.42.2.2 orientation

`quantity<si::plane_angle, double> oCpt::components::sensors::KalmanIMU::ReturnValue:::orientation[3]`

Definition at line 235 of file KalmanIMU.h.

6.42.2.3 orientation_change

`quantity<angular_velocity_dimension, double> oCpt::components::sensors::KalmanIMU::ReturnValue:::orientation_change[3]`

Definition at line 236 of file KalmanIMU.h.

6.42.2.4 position

`quantity<si::length, double> oCpt::components::sensors::KalmanIMU::ReturnValue:::position[2]`

Definition at line 232 of file KalmanIMU.h.

6.42.2.5 velocity

`quantity<si::velocity, double> oCpt::components::sensors::KalmanIMU::ReturnValue:::velocity[2]`

Definition at line 233 of file KalmanIMU.h.

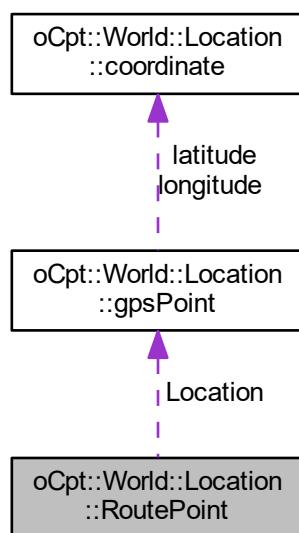
The documentation for this struct was generated from the following file:

- /projects/mti/ohCaptain/ohCaptain/include/Sensors/[KalmanIMU.h](#)

6.43 oCpt::World::Location::RoutePoint Struct Reference

#include <World.h>

Collaboration diagram for oCpt::World::Location::RoutePoint:



Public Types

- `typedef boost::shared_ptr<RoutePoint> ptr`

Public Attributes

- `Time::timepoint_t TimePoint`
- `gpsPoint_t Location`

6.43.1 Detailed Description

Definition at line 153 of file World.h.

6.43.2 Member Typedef Documentation

6.43.2.1 `ptr`

```
typedef boost::shared_ptr<RoutePoint> oCpt::World::Location::RoutePoint::ptr
```

Definition at line 154 of file World.h.

6.43.3 Member Data Documentation

6.43.3.1 `Location`

```
gpsPoint_t oCpt::World::Location::RoutePoint::Location
```

Definition at line 156 of file World.h.

6.43.3.2 `TimePoint`

```
Time::timepoint_t oCpt::World::Location::RoutePoint::TimePoint
```

Definition at line 155 of file World.h.

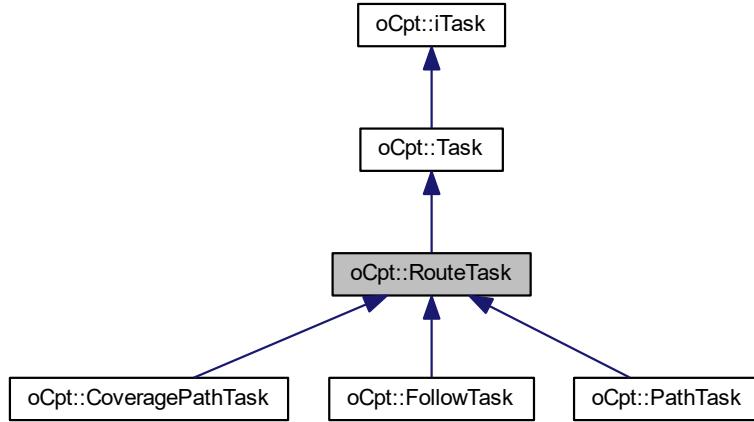
The documentation for this struct was generated from the following file:

- `/projects/mti/ohCaptain/ohCaptain/include/Core/World.h`

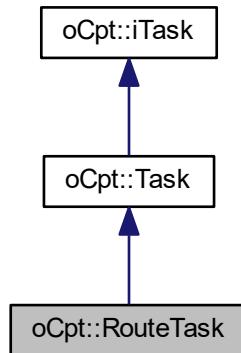
6.44 oCpt::RouteTask Class Reference

```
#include <Task.h>
```

Inheritance diagram for oCpt::RouteTask:



Collaboration diagram for oCpt::RouteTask:



Public Member Functions

- `RouteTask (Vessel::ptr vessel, bool concurrent=false)`
- virtual `~RouteTask ()`

Additional Inherited Members

6.44.1 Detailed Description

An object representing route related tasks

Definition at line 148 of file Task.h.

6.44.2 Constructor & Destructor Documentation

6.44.2.1 RouteTask()

```
oCpt::RouteTask::RouteTask (
    Vessel::ptr vessel,
    bool concurrent = false )
```

Constructor of the interface

Returns

Definition at line 41 of file Task.cpp.

References oCpt::Task::__typeof.

6.44.2.2 ~RouteTask()

```
oCpt::RouteTask::~RouteTask ( ) [virtual]
```

The deconstructor

Definition at line 45 of file Task.cpp.

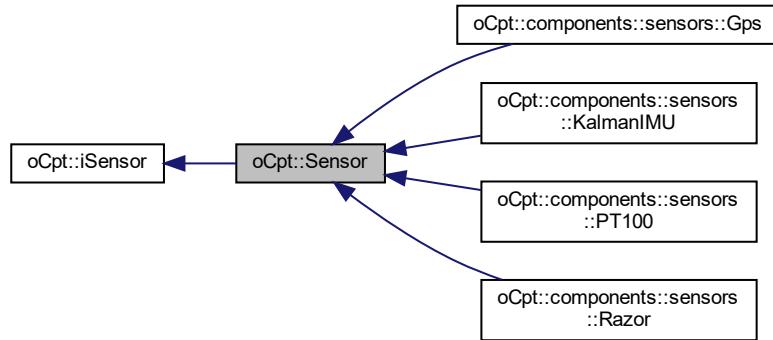
The documentation for this class was generated from the following files:

- /projects/mti/ohCaptain/ohCaptain/include/Core/[Task.h](#)
- /projects/mti/ohCaptain/ohCaptain/src/Core/[Task.cpp](#)

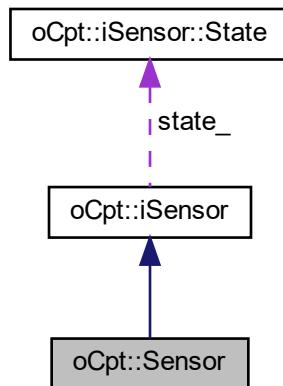
6.45 oCpt::Sensor Class Reference

```
#include <Sensor.h>
```

Inheritance diagram for oCpt::Sensor:



Collaboration diagram for oCpt::Sensor:



Public Member Functions

- `Sensor (iController::ptr controller, World::ptr world, std::string id, std::string typeOfSensor)`
- virtual `~Sensor ()` override
- virtual void `updateSensor ()` override
- virtual void `run ()` override
- virtual void `stop ()` override
- virtual void `init ()` override
- virtual void `setIoservice (boost::shared_ptr< boost::asio::io_service > ioservice)` override

Additional Inherited Members

6.45.1 Detailed Description

Implementation of the [iSensor](#) interface

Definition at line 152 of file Sensor.h.

6.45.2 Constructor & Destructor Documentation

6.45.2.1 Sensor()

```
oCpt::Sensor::Sensor (
    iController::ptr controller,
    World::ptr world,
    std::string id,
    std::string typeOfSensor )
```

Constructor of [Sensor](#)

Parameters

<i>controller</i>	a shared_ptr of the controller where the sensor is hooked to
<i>world</i>	a shared_ptr of the world in which the vessel operates
<i>id</i>	a identifying name of the sensor
<i>typeOfSensor</i>	a identifying category for the sensor

Definition at line 57 of file Sensor.cpp.

6.45.2.2 ~Sensor()

```
oCpt::Sensor::~Sensor ( ) [override], [virtual]
```

Deconstructor of the [Sensor](#) class

Definition at line 62 of file Sensor.cpp.

6.45.3 Member Function Documentation

6.45.3.1 init()

```
void oCpt::Sensor::init ( ) [override], [virtual]
```

Initialize the sensor

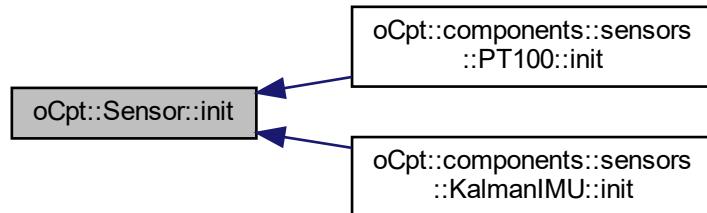
Implements [oCpt::iSensor](#).

Reimplemented in [oCpt::components::sensors::KalmanIMU](#), [oCpt::components::sensors::Razor](#), and [oCpt::components::sensors::PT100](#).

Definition at line 78 of file Sensor.cpp.

Referenced by [oCpt::components::sensors::PT100::init\(\)](#), and [oCpt::components::sensors::KalmanIMU::init\(\)](#).

Here is the caller graph for this function:



6.45.3.2 run()

`void oCpt::Sensor::run () [override], [virtual]`

virtual function starting the run service for the IO

Implements [oCpt::iSensor](#).

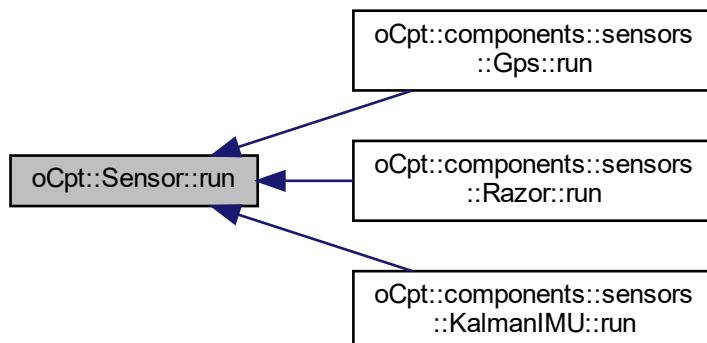
Reimplemented in [oCpt::components::sensors::KalmanIMU](#), [oCpt::components::sensors::Razor](#), [oCpt::components::sensors::PT100](#), and [oCpt::components::sensors::Gps](#).

Definition at line 70 of file Sensor.cpp.

References [oCpt::iSensor::sensorRunning_](#).

Referenced by [oCpt::components::sensors::Gps::run\(\)](#), [oCpt::components::sensors::Razor::run\(\)](#), and [oCpt::components::sensors::KalmanIMU::run\(\)](#).

Here is the caller graph for this function:



6.45.3.3 setIoservice()

```
void oCpt::Sensor::setIoservice (
    boost::shared_ptr< boost::asio::io_service > ioservice ) [override], [virtual]
```

Setting the used Asynchronous Input Output service

Parameters

<i>ioservice</i>	ASIO IO service, which handles the async calls from multiple sensors
------------------	--

Implements [oCpt::iSensor](#).

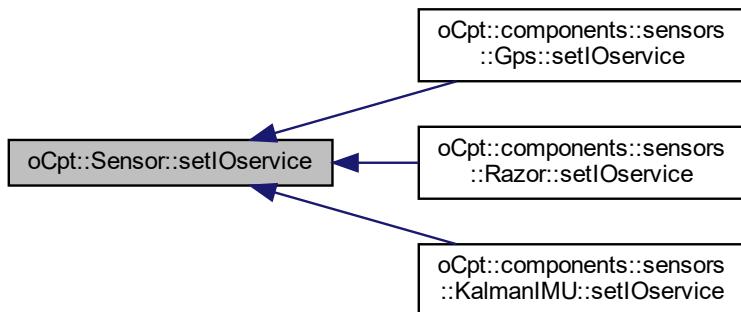
Reimplemented in [oCpt::components::sensors::KalmanIMU](#), [oCpt::components::sensors::Razor](#), and [oCpt::components::sensors::Gps](#).

Definition at line 82 of file Sensor.cpp.

References [oCpt::iSensor::ioservice_](#).

Referenced by [oCpt::components::sensors::Gps::setIoservice\(\)](#), [oCpt::components::sensors::Razor::setIoservice\(\)](#), and [oCpt::components::sensors::KalmanIMU::setIoservice\(\)](#).

Here is the caller graph for this function:



6.45.3.4 stop()

```
void oCpt::Sensor::stop ( ) [override], [virtual]
```

virtual function stopping the run

Implements [oCpt::iSensor](#).

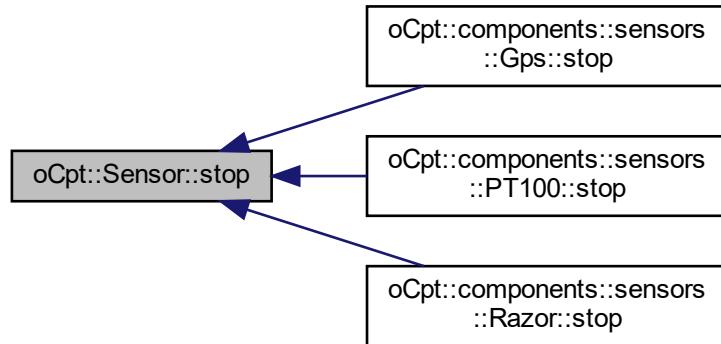
Reimplemented in [oCpt::components::sensors::Razor](#), [oCpt::components::sensors::PT100](#), and [oCpt::components::sensors::Gps](#).

Definition at line 74 of file Sensor.cpp.

References oCpt::iSensor::sensorRunning_.

Referenced by oCpt::components::sensors::Gps::stop(), oCpt::components::sensors::PT100::stop(), and oCpt::components::sensors::Razor::stop().

Here is the caller graph for this function:



6.45.3.5 updateSensor()

`void oCpt::Sensor::updateSensor () [override], [virtual]`

virtual function which performs a sensor update, obtaining a new value and sending a signal afterwards

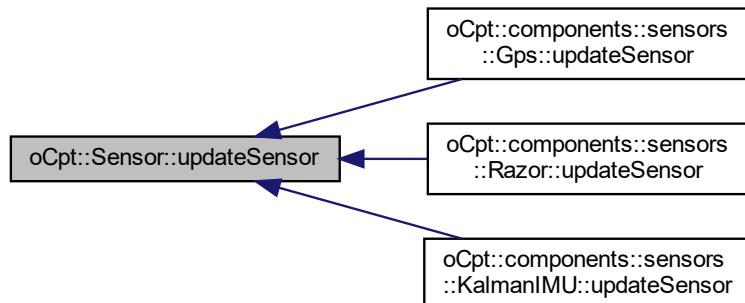
Implements [oCpt::iSensor](#).

Reimplemented in [oCpt::components::sensors::KalmanIMU](#), [oCpt::components::sensors::Razor](#), [oCpt::components::sensors::PT100](#), and [oCpt::components::sensors::Gps](#).

Definition at line 66 of file Sensor.cpp.

Referenced by [oCpt::components::sensors::Gps::updateSensor\(\)](#), [oCpt::components::sensors::Razor::updateSensor\(\)](#), and [oCpt::components::sensors::KalmanIMU::updateSensor\(\)](#).

Here is the caller graph for this function:



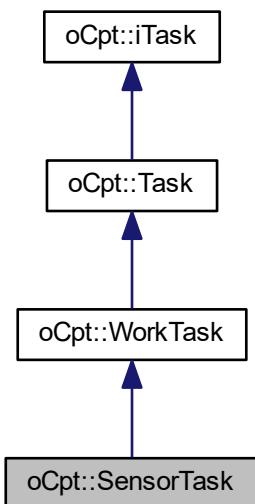
The documentation for this class was generated from the following files:

- /projects/mti/ohCaptain/ohCaptain/include/Core/Sensor.h
- /projects/mti/ohCaptain/ohCaptain/src/Core/Sensor.cpp

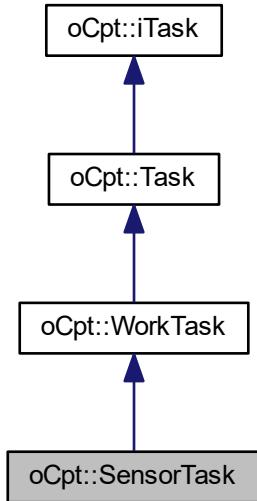
6.46 oCpt::SensorTask Class Reference

```
#include <Task.h>
```

Inheritance diagram for oCpt::SensorTask:



Collaboration diagram for oCpt::SensorTask:



Public Member Functions

- [SensorTask \(Vessel::ptr vessel, bool concurrent=true\)](#)
- virtual [~SensorTask \(\)](#)

Additional Inherited Members

6.46.1 Detailed Description

Definition at line 303 of file Task.h.

6.46.2 Constructor & Destructor Documentation

6.46.2.1 SensorTask()

```
oCpt::SensorTask::SensorTask (
    Vessel::ptr vessel,
    bool concurrent = true )
```

Definition at line 73 of file Task.cpp.

6.46.2.2 ~SensorTask()

```
oCpt::SensorTask::~SensorTask ( ) [virtual]
```

Definition at line 75 of file Task.cpp.

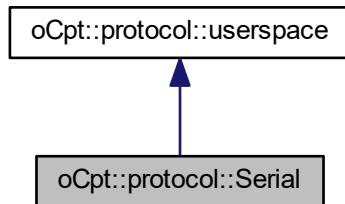
The documentation for this class was generated from the following files:

- /projects/mti/ohCaptain/ohCaptain/include/Core/[Task.h](#)
- /projects/mti/ohCaptain/ohCaptain/src/Core/[Task.cpp](#)

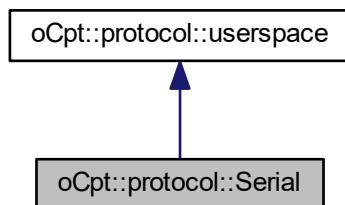
6.47 oCpt::protocol::Serial Class Reference

```
#include <Controller.h>
```

Inheritance diagram for oCpt::protocol::Serial:



Collaboration diagram for oCpt::protocol::Serial:



Public Types

- `typedef boost::shared_ptr< Serial > ptr`
- `typedef std::function< void(const unsigned char *, size_t)> cb_func`
- `typedef boost::asio::serial_port_base::parity parity_t`
- `typedef boost::asio::serial_port_base::character_size character_size_t`
- `typedef boost::asio::serial_port_base::flow_control flow_control_t`
- `typedef boost::asio::serial_port_base::stop_bits stop_bits_t`
- `typedef boost::shared_ptr< boost::asio::io_service > io_service_t`
- `typedef boost::asio::serial_port serialport_t`
- `typedef boost::signals2::signal< void()> signal_t`

Public Member Functions

- `Serial (const std::string &device, unsigned int baudrate, io_service_t ioservice=io_service_t(new boost::asio::io_service()), parity_t parity=parity_t(parity_t::none), character_size_t csizesize_t(8), flow_control_t flow=flow_control_t(flow_control_t::none), stop_bits_t stop=stop_bits_t(stop_bits_t::one), unsigned int maxreadlenthg=4096)`
- `void open ()`
- `void start ()`
- `bool isOpen ()`
- `void close ()`
- `bool write (const std::string &msg)`
- `bool write (const std::vector< unsigned char > &data)`
- `void setReadCallback (cb_func cb_function)`
- `void setIoservice (boost::shared_ptr< boost::asio::io_service > io_ptr)`
- `std::deque< std::string > * getReturnMsgQueue ()`
- `std::string readFiFoMsg ()`
- `void setMaxReadLength (unsigned int maxReadLength)`

Public Attributes

- `signal_t msgRecievedSig`

Protected Member Functions

- `void internalCallback (const unsigned char *data, size_t size)`
- `void closeCallback (const boost::system::error_code &error)`
- `void readComplete (const boost::system::error_code &error, size_t bytes_transferred)`
- `void writeCallback (const std::vector< unsigned char > &msg)`
- `void writeStart ()`
- `void writeComplete (const boost::system::error_code &error)`
- `void ReadStart ()`

Protected Attributes

- unsigned int `maxReadLength_`
- std::deque< std::vector< unsigned char > > `msgQueue_`
- std::deque< std::string > `returnMsgQueue_`
- std::string `msg_`
- std::string `receivedMsg_`
- unsigned char `read_msg` [4096]
- `cb_func callback_`
- `io_service_t ioservice_`
- std::string `device_`
- unsigned int `baudrate_`
- `parity_t parity_`
- `character_size_t csize_`
- `flow_control_t flow_`
- `stop_bits_t stop_`
- `serialport_t serialport_`
- bool `firstMsg` = true

6.47.1 Detailed Description

Communication via the serial port, using an Asynchronous Input Output setup, provided by Boost. All communication is handled on the background via a `io_service`. When data is received a callback function is called. This can either be an external function or an internal one, which sends a signal for each new line. The lines can then be read using a FiFo function.

Definition at line 390 of file Controller.h.

6.47.2 Member Typedef Documentation

6.47.2.1 `cb_func`

```
typedef std::function<void(const unsigned char *, size_t)> oCpt::protocol::Serial::cb_func
```

Definition at line 394 of file Controller.h.

6.47.2.2 `character_size_t`

```
typedef boost::asio::serial_port_base::character_size oCpt::protocol::Serial::character_size_t
```

Definition at line 396 of file Controller.h.

6.47.2.3 `flow_control_t`

```
typedef boost::asio::serial_port_base::flow_control oCpt::protocol::Serial::flow_control_t
```

Definition at line 397 of file Controller.h.

6.47.2.4 io_service_t

```
typedef boost::shared_ptr<boost::asio::io_service> oCpt::protocol::Serial::io_service_t
```

Definition at line 399 of file Controller.h.

6.47.2.5 parity_t

```
typedef boost::asio::serial_port_base::parity oCpt::protocol::Serial::parity_t
```

Definition at line 395 of file Controller.h.

6.47.2.6 ptr

```
typedef boost::shared_ptr<Serial> oCpt::protocol::Serial::ptr
```

Definition at line 392 of file Controller.h.

6.47.2.7 serialport_t

```
typedef boost::asio::serial_port oCpt::protocol::Serial::serialport_t
```

Definition at line 400 of file Controller.h.

6.47.2.8 signal_t

```
typedef boost::signals2::signal<void()> oCpt::protocol::Serial::signal_t
```

Definition at line 401 of file Controller.h.

6.47.2.9 stop_bits_t

```
typedef boost::asio::serial_port_base::stop_bits oCpt::protocol::Serial::stop_bits_t
```

Definition at line 398 of file Controller.h.

6.47.3 Constructor & Destructor Documentation

6.47.3.1 Serial()

```
oCpt::protocol::Serial::Serial (
    const std::string & device,
    unsigned int baudrate,
    io_service_t ioservice = io_service_t(new boost::asio::io_service()),
    Serial::parity_t parity = parity_t(parity_t::none),
    Serial::character_size_t csize = character_size_t(8),
    Serial::flow_control_t flow = flow_control_t(flow_control_t::none),
    Serial::stop_bits_t stop = stop_bits_t(stop_bits_t::one),
    unsigned int maxreadlentgh = 4096 )
```

Constructor of the [Serial](#) class

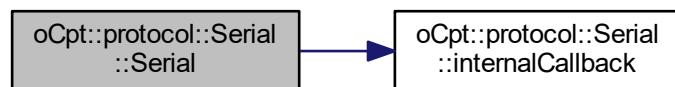
Parameters

<i>device</i>	a string representing the device path eq. /dev/tty0
<i>baudrate</i>	the baudrate of the device eq. 9600, 57600, 115200
<i>ioservice</i>	the io service to be used standard it's a new service
<i>parity</i>	the parity of the Serial port with a standard parity of Parity_t::none
<i>csize</i>	The character_size with a standard value of 8
<i>flow</i>	The flow control of the device with a standard value of flow_control_t::none
<i>stop</i>	The stop bit of the device with a standard value of stop_bits_t::none
<i>maxreadlenth</i>	the Maximum buffer with a standard value of 4096

Definition at line 99 of file Controller.cpp.

References [callback_](#), and [internalCallback\(\)](#).

Here is the call graph for this function:



6.47.4 Member Function Documentation

6.47.4.1 close()

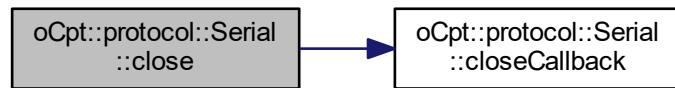
```
void oCpt::protocol::Serial::close( )
```

Closes the port when it is open

Definition at line 142 of file Controller.cpp.

References [closeCallback\(\)](#), [ioservice_](#), and [serialport_](#).

Here is the call graph for this function:



6.47.4.2 closeCallback()

```
void oCpt::protocol::Serial::closeCallback (
    const boost::system::error_code & error ) [protected]
```

The callback function which is called after the port is closed

Parameters

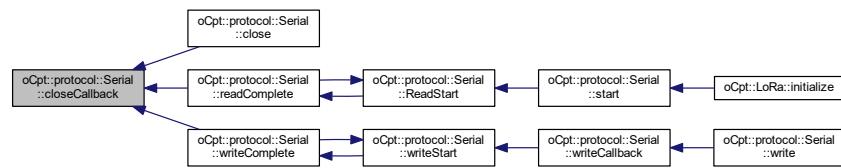
<i>error</i>	an past trhough boost::system::error_code
--------------	---

Definition at line 159 of file Controller.cpp.

References serialport_.

Referenced by close(), readComplete(), and writeComplete().

Here is the caller graph for this function:

**6.47.4.3 getReturnMsgQueue()**

```
std::deque< std::string > * oCpt::protocol::Serial::getReturnMsgQueue ( )
```

Get the complete returnMsg que

Returns

a deque with all the return lines

Definition at line 197 of file Controller.cpp.

References returnMsgQueue_.

6.47.4.4 internalCallback()

```
void oCpt::protocol::Serial::internalCallback (
    const unsigned char * data,
    size_t size ) [protected]
```

The internal callback function, which handles messages longer then maxreadlentgh and splits the message with

Parameters

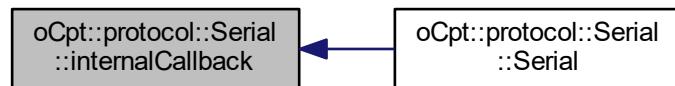
<i>data</i>	the buffer obtained by the serial port
<i>size</i>	the size obtained

Definition at line 208 of file Controller.cpp.

References msgReceivedSig, receivedMsg_, and returnMsgQueue_.

Referenced by Serial().

Here is the caller graph for this function:



6.47.4.5 isOpen()

```
bool oCpt::protocol::Serial::isOpen( )
```

Checks if the port is open

Returns

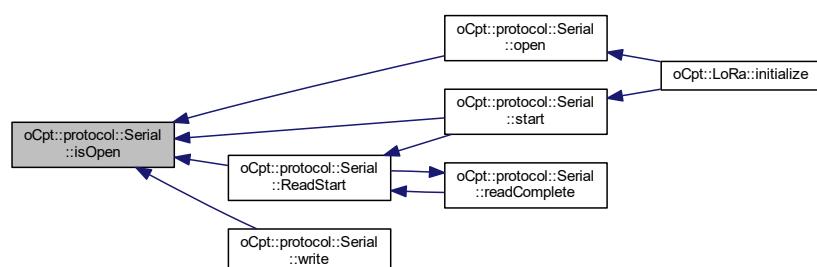
either true or false

Definition at line 138 of file Controller.cpp.

References serialport_.

Referenced by open(), ReadStart(), start(), and write().

Here is the caller graph for this function:



6.47.4.6 open()

```
void oCpt::protocol::Serial::open ( )
```

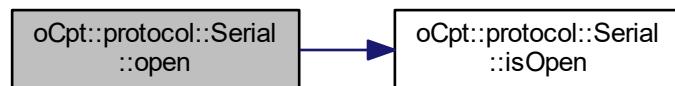
Open teh serial port

Definition at line 118 of file Controller.cpp.

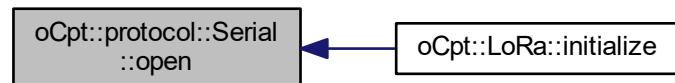
References baudrate_, csize_, device_, boost::units::constants::e, flow_, ioservice_, isOpen(), parity_, serialport_, and stop_.

Referenced by oCpt::LoRa::initialize().

Here is the call graph for this function:



Here is the caller graph for this function:



6.47.4.7 readComplete()

```
void oCpt::protocol::Serial::readComplete (
    const boost::system::error_code & error,
    size_t bytes_transferred ) [protected]
```

The callbackfunction to be performed when reading is complete

Parameters

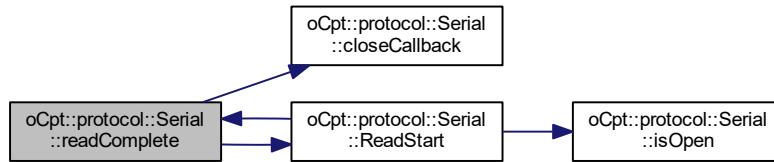
<i>error</i>	boost::system::error_code if an error is presented
<i>bytes_transferred</i>	number of bytes that are transferred

Definition at line 188 of file Controller.cpp.

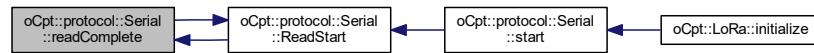
References callback_, closeCallback(), read_msg, and ReadStart().

Referenced by ReadStart().

Here is the call graph for this function:



Here is the caller graph for this function:



6.47.4.8 readFiFoMsg()

```
std::string oCpt::protocol::Serial::readFiFoMsg( )
```

Gets the first received message, which is then removed from the queue

Returns

Definition at line 201 of file Controller.cpp.

References returnMsgQueue_.

Referenced by oCpt::LoRa::messageReceived().

Here is the caller graph for this function:



6.47.4.9 ReadStart()

```
void oCpt::protocol::Serial::ReadStart ( ) [protected]
```

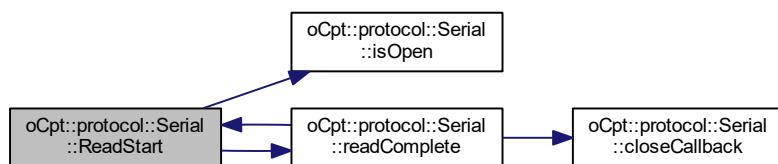
Start the reading process

Definition at line 178 of file Controller.cpp.

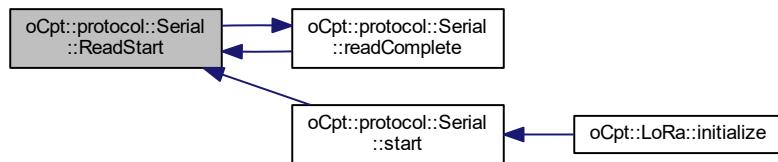
References isOpen(), maxReadLength_, read_msg, readComplete(), and serialport_.

Referenced by readComplete(), and start().

Here is the call graph for this function:



Here is the caller graph for this function:



6.47.4.10 setIoservice()

```
void oCpt::protocol::Serial::setIoservice (
    boost::shared_ptr< boost::asio::io_service > io_ptr )
```

set a new IO service

Parameters

<i>io_ptr</i>	a shared_ptr to the new IO service
---------------	------------------------------------

Definition at line 155 of file Controller.cpp.

References ioservice_.

6.47.4.11 setMaxReadLength()

```
void oCpt::protocol::Serial::setMaxReadLength (
    unsigned int maxReadLength )
```

Set the maximum buffer of the [Serial](#) class

Parameters

<i>maxReadLength</i>	the number of bytes
----------------------	---------------------

Definition at line 287 of file Controller.cpp.

References [maxReadLength_](#).

6.47.4.12 setReadCallback()

```
void oCpt::protocol::Serial::setReadCallback (
    cb_func cb_function )
```

Set a new callback function

Parameters

<i>cb_function</i>	the callback function
--------------------	-----------------------

Definition at line 151 of file Controller.cpp.

References [callback_](#).

6.47.4.13 start()

```
void oCpt::protocol::Serial::start ( )
```

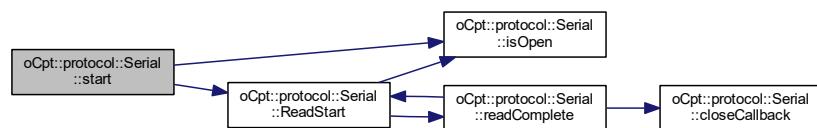
Start the [io_service](#) on a sepearte thread

Definition at line 166 of file Controller.cpp.

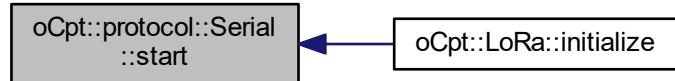
References [ioservice_](#), [isOpen\(\)](#), and [ReadStart\(\)](#).

Referenced by [oCpt::LoRa::initialize\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



6.47.4.14 write() [1/2]

```
bool oCpt::protocol::Serial::write (
    const std::string & msg )
```

Write a message to the port

Parameters

<i>msg</i>	a string with the payload
------------	---------------------------

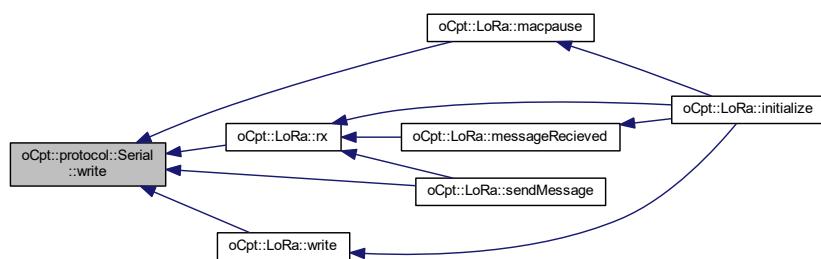
Returns

either true or false depending if writing was successful or not

Definition at line 238 of file Controller.cpp.

Referenced by oCpt::LoRa::macpause(), oCpt::LoRa::rx(), oCpt::LoRa::sendMessage(), and oCpt::LoRa::write().

Here is the caller graph for this function:



6.47.4.15 write() [2/2]

```
bool oCpt::protocol::Serial::write (
    const std::vector< unsigned char > & data )
```

Write a message as a vector of unsigned chars

Parameters

<i>data</i>	the message to be send
-------------	------------------------

Returns

either true or false depending if writing was successful or not

Definition at line 244 of file Controller.cpp.

References ioservice_, isOpen(), and writeCallback().

Here is the call graph for this function:

**6.47.4.16 writeCallback()**

```
void oCpt::protocol::Serial::writeCallback (
    const std::vector< unsigned char > & msg ) [protected]
```

When the writing is finished call this function, which will write the next message if present

Parameters

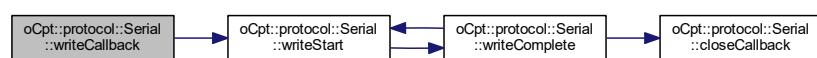
<i>msg</i>	the message to write as an vector of unsigned char
------------	--

Definition at line 256 of file Controller.cpp.

References msgQueue_, and writeStart().

Referenced by `write()`.

Here is the call graph for this function:



Here is the caller graph for this function:



6.47.4.17 writeComplete()

```
void oCpt::protocol::Serial::writeComplete (
    const boost::system::error_code & error ) [protected]
```

restart the write process when the previous write is finished

Parameters

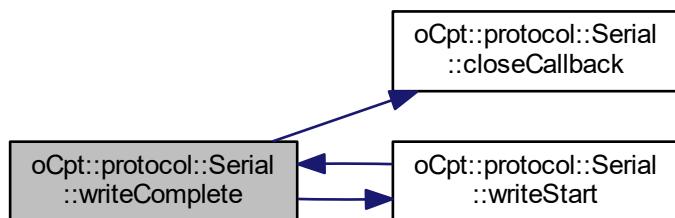
error

Definition at line 275 of file Controller.cpp.

References closeCallback(), msgQueue_, and writeStart().

Referenced by writeStart().

Here is the call graph for this function:



Here is the caller graph for this function:



6.47.4.18 writeStart()

```
void oCpt::protocol::Serial::writeStart ( ) [protected]
```

Start with the write sequence

Definition at line 264 of file Controller.cpp.

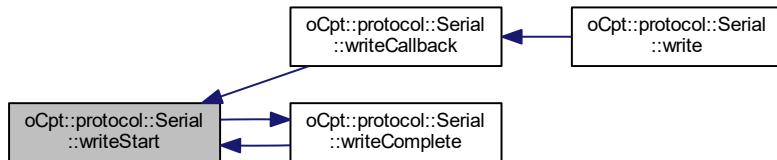
References msgQueue_, serialport_, and writeComplete().

Referenced by writeCallback(), and writeComplete().

Here is the call graph for this function:



Here is the caller graph for this function:



6.47.5 Member Data Documentation

6.47.5.1 baudrate_

```
unsigned int oCpt::protocol::Serial::baudrate_ [protected]
```

Definition at line 546 of file Controller.h.

Referenced by open().

6.47.5.2 callback_

```
cb_func oCpt::protocol::Serial::callback_ [protected]
```

Definition at line 543 of file Controller.h.

Referenced by readComplete(), Serial(), and setReadCallback().

6.47.5.3 `csize_`

```
character_size_t oCpt::protocol::Serial::csize_ [protected]
```

Definition at line 548 of file Controller.h.

Referenced by `open()`.

6.47.5.4 `device_`

```
std::string oCpt::protocol::Serial::device_ [protected]
```

Definition at line 545 of file Controller.h.

Referenced by `open()`.

6.47.5.5 `firstMsg`

```
bool oCpt::protocol::Serial::firstMsg = true [protected]
```

Definition at line 552 of file Controller.h.

6.47.5.6 `flow_`

```
flow_control_t oCpt::protocol::Serial::flow_ [protected]
```

Definition at line 549 of file Controller.h.

Referenced by `open()`.

6.47.5.7 `ioservice_`

```
io_service_t oCpt::protocol::Serial::ioservice_ [protected]
```

Definition at line 544 of file Controller.h.

Referenced by `close()`, `open()`, `setIoservice()`, `start()`, and `write()`.

6.47.5.8 `maxReadLength_`

```
unsigned int oCpt::protocol::Serial::maxReadLength_ [protected]
```

Definition at line 529 of file Controller.h.

Referenced by `ReadStart()`, and `setMaxReadLength()`.

6.47.5.9 msg_

```
std::string oCpt::protocol::Serial::msg_ [protected]
```

Definition at line 540 of file Controller.h.

6.47.5.10 msgQueue_

```
std::deque<std::vector<unsigned char>> oCpt::protocol::Serial::msgQueue_ [protected]
```

Definition at line 538 of file Controller.h.

Referenced by writeCallback(), writeComplete(), and writeStart().

6.47.5.11 msgRecievedSig

```
signal_t oCpt::protocol::Serial::msgRecievedSig
```

The signal which is send when a new line has been received or the buffer is full

Definition at line 484 of file Controller.h.

Referenced by oCpt::LoRa::initialize(), and internalCallback().

6.47.5.12 parity_

```
parity_t oCpt::protocol::Serial::parity_ [protected]
```

Definition at line 547 of file Controller.h.

Referenced by open().

6.47.5.13 read_msg

```
unsigned char oCpt::protocol::Serial::read_msg[4096] [protected]
```

Definition at line 542 of file Controller.h.

Referenced by readComplete(), and ReadStart().

6.47.5.14 receivedMsg_

```
std::string oCpt::protocol::Serial::receivedMsg_ [protected]
```

Definition at line 541 of file Controller.h.

Referenced by internalCallback().

6.47.5.15 returnMsgQueue_

```
std::deque<std::string> oCpt::protocol::Serial::returnMsgQueue_ [protected]
```

Definition at line 539 of file Controller.h.

Referenced by `getReturnMsgQueue()`, `internalCallback()`, and `readFiFoMsg()`.

6.47.5.16 serialport_

```
serialport_t oCpt::protocol::Serial::serialport_ [protected]
```

Definition at line 551 of file Controller.h.

Referenced by `close()`, `closeCallback()`, `isOpen()`, `open()`, `ReadStart()`, and `writeStart()`.

6.47.5.17 stop_

```
stop_bits_t oCpt::protocol::Serial::stop_ [protected]
```

Definition at line 550 of file Controller.h.

Referenced by `open()`.

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

- /projects/mti/ohCaptain/ohCaptain/include/Core/[Controller.h](#)
- /projects/mti/ohCaptain/ohCaptain/src/Core/[Controller.cpp](#)

6.48 oCpt::iSensor::State Struct Reference

```
#include <Sensor.h>
```

Public Attributes

- `generic_t Value`
- `World::Time::timepoint_t Stamp`

6.48.1 Detailed Description

Definition at line 35 of file Sensor.h.

6.48.2 Member Data Documentation

6.48.2.1 Stamp

```
World::Time::timepoint_t oCpt::iSensor::State::Stamp
```

Definition at line 37 of file Sensor.h.

Referenced by oCpt::components::sensors::Gps::interpretMsg(), oCpt::components::sensors::Razor::msgHandler(), oCpt::components::sensors::Razor::Razor(), oCpt::components::sensors::PT100::updateSensor(), and oCpt::components::sensors::Razor::updateSensor().

6.48.2.2 Value

```
generic_t oCpt::iSensor::State::Value
```

Definition at line 36 of file Sensor.h.

Referenced by oCpt::components::sensors::Gps::interpretMsg(), oCpt::iSensor::iSensor(), oCpt::components::sensors::Razor::msgHandler(), oCpt::components::sensors::Razor::Razor(), and oCpt::components::sensors::PT100::updateSensor().

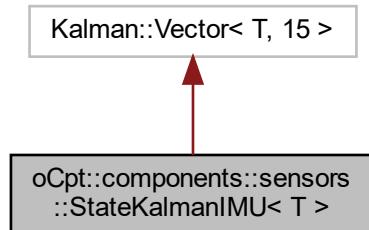
The documentation for this struct was generated from the following file:

- /projects/mti/ohCaptain/ohCaptain/include/Core/Sensor.h

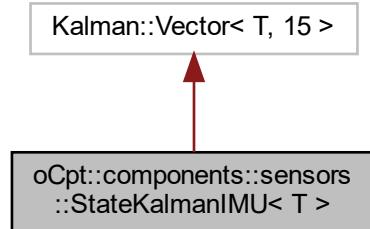
6.49 oCpt::components::sensors::StateKalmanIMU< T > Class Template Reference

```
#include <KalmanIMU.h>
```

Inheritance diagram for oCpt::components::sensors::StateKalmanIMU< T >:



Collaboration diagram for oCpt::components::sensors::StateKalmanIMU< T >:



Public Types

- `typedef Kalman::Vector< T, 15 > Base`

Public Member Functions

- `StateKalmanIMU (void)`
- `template<typename OtherDerived>`
`StateKalmanIMU (const Eigen::MatrixBase< OtherDerived > &other)`
- `template<typename OtherDerived>`
`StateKalmanIMU & operator= (const Eigen::MatrixBase< OtherDerived > &other)`
- `T pos_x () const`
- `T pos_y () const`
- `T pos_z () const`
- `T vel_x () const`
- `T vel_y () const`
- `T vel_z () const`
- `T acc_x () const`
- `T acc_y () const`
- `T acc_z () const`
- `T theta () const`
- `T psi () const`
- `T phi () const`
- `T thetaPrime () const`
- `T psiPrime () const`
- `T phiPrime () const`
- `T & pos_x ()`
- `T & pos_y ()`
- `T & pos_z ()`
- `T & vel_x ()`
- `T & vel_y ()`
- `T & vel_z ()`
- `T & acc_x ()`
- `T & acc_y ()`
- `T & acc_z ()`
- `T & theta ()`

- T & `psi ()`
- T & `phi ()`
- T & `thetaPrime ()`
- T & `psiPrime ()`
- T & `phiPrime ()`

Static Public Attributes

- static constexpr size_t `posX` = 0
- static constexpr size_t `posY` = 1
- static constexpr size_t `posZ` = 2
- static constexpr size_t `velX` = 3
- static constexpr size_t `velY` = 4
- static constexpr size_t `velZ` = 5
- static constexpr size_t `accX` = 6
- static constexpr size_t `accY` = 7
- static constexpr size_t `accZ` = 8
- static constexpr size_t `Theta` = 9
- static constexpr size_t `Psi` = 10
- static constexpr size_t `Phi` = 11
- static constexpr size_t `ThetaPrime` = 12
- static constexpr size_t `PsiPrime` = 13
- static constexpr size_t `PhiPrime` = 14

6.49.1 Detailed Description

```
template<typename T>
class oCpt::components::sensors::StateKalmanIMU< T >
```

Definition at line 18 of file KalmanIMU.h.

6.49.2 Member Typedef Documentation

6.49.2.1 Base

```
template<typename T >
typedef Kalman::Vector< T , 15 > oCpt::components::sensors::StateKalmanIMU< T >::Base
```

Definition at line 20 of file KalmanIMU.h.

6.49.3 Constructor & Destructor Documentation

6.49.3.1 StateKalmanIMU() [1/2]

```
template<typename T >
oCpt::components::sensors::StateKalmanIMU< T >::StateKalmanIMU (
    void ) [inline]
```

Definition at line 20 of file KalmanIMU.h.

6.49.3.2 StateKalmanIMU() [2/2]

```
template<typename T >
template<typename OtherDerived >
oCpt::components::sensors::StateKalmanIMU< T >::StateKalmanIMU (
    const Eigen::MatrixBase< OtherDerived > & other ) [inline]
```

Definition at line 20 of file KalmanIMU.h.

6.49.4 Member Function Documentation

6.49.4.1 acc_x() [1/2]

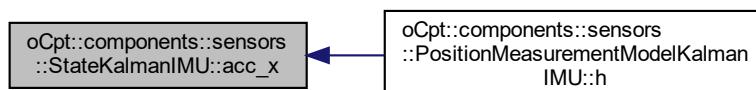
```
template<typename T >
T oCpt::components::sensors::StateKalmanIMU< T >::acc_x ( ) const [inline]
```

Definition at line 50 of file KalmanIMU.h.

References oCpt::components::sensors::StateKalmanIMU< T >::accX.

Referenced by oCpt::components::sensors::PositionMeasurementModelKalmanIMU< T, CovarianceBase >::h().

Here is the caller graph for this function:



6.49.4.2 acc_x() [2/2]

```
template<typename T >
T& oCpt::components::sensors::StateKalmanIMU< T >::acc_x ( ) [inline]
```

Definition at line 80 of file KalmanIMU.h.

References oCpt::components::sensors::StateKalmanIMU< T >::accX.

6.49.4.3 acc_y() [1/2]

```
template<typename T >
T oCpt::components::sensors::StateKalmanIMU< T >::acc_y ( ) const [inline]
```

Definition at line 52 of file KalmanIMU.h.

References oCpt::components::sensors::StateKalmanIMU< T >::accY.

Referenced by oCpt::components::sensors::PositionMeasurementModelKalmanIMU< T, CovarianceBase >::h().

Here is the caller graph for this function:



6.49.4.4 acc_y() [2/2]

```
template<typename T >
T& oCpt::components::sensors::StateKalmanIMU< T >::acc_y ( ) [inline]
```

Definition at line 82 of file KalmanIMU.h.

References oCpt::components::sensors::StateKalmanIMU< T >::accY.

6.49.4.5 acc_z() [1/2]

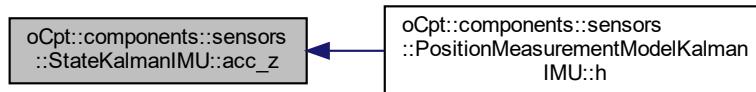
```
template<typename T >
T oCpt::components::sensors::StateKalmanIMU< T >::acc_z ( ) const [inline]
```

Definition at line 54 of file KalmanIMU.h.

References oCpt::components::sensors::StateKalmanIMU< T >::accZ.

Referenced by oCpt::components::sensors::PositionMeasurementModelKalmanIMU< T, CovarianceBase >::h().

Here is the caller graph for this function:



6.49.4.6 acc_z() [2/2]

```
template<typename T >
T& oCpt::components::sensors::StateKalmanIMU< T >::acc_z ( ) [inline]
```

Definition at line 84 of file KalmanIMU.h.

References oCpt::components::sensors::StateKalmanIMU< T >::accZ.

6.49.4.7 operator=()

```
template<typename T >
template<typename OtherDerived >
StateKalmanIMU& oCpt::components::sensors::StateKalmanIMU< T >::operator= (
    const Eigen::MatrixBase< OtherDerived > & other) [inline]
```

Definition at line 20 of file KalmanIMU.h.

6.49.4.8 phi() [1/2]

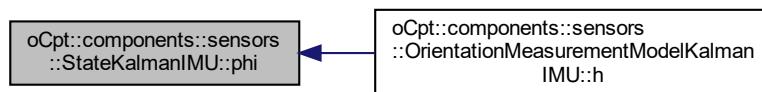
```
template<typename T >
T oCpt::components::sensors::StateKalmanIMU< T >::phi ( ) const [inline]
```

Definition at line 60 of file KalmanIMU.h.

References oCpt::components::sensors::StateKalmanIMU< T >::Phi.

Referenced by oCpt::components::sensors::OrientationMeasurementModelKalmanIMU< T, CovarianceBase >::h().

Here is the caller graph for this function:



6.49.4.9 phi() [2/2]

```
template<typename T >
T& oCpt::components::sensors::StateKalmanIMU< T >::phi ( ) [inline]
```

Definition at line 90 of file KalmanIMU.h.

References oCpt::components::sensors::StateKalmanIMU< T >::Phi.

6.49.4.10 phiPrime() [1/2]

```
template<typename T >
T oCpt::components::sensors::StateKalmanIMU< T >::phiPrime ( ) const [inline]
```

Definition at line 66 of file KalmanIMU.h.

References oCpt::components::sensors::StateKalmanIMU< T >::PhiPrime.

Referenced by oCpt::components::sensors::OrientationMeasurementModelKalmanIMU< T, CovarianceBase >::h().

Here is the caller graph for this function:



6.49.4.11 phiPrime() [2/2]

```
template<typename T >
T& oCpt::components::sensors::StateKalmanIMU< T >::phiPrime ( ) [inline]
```

Definition at line 96 of file KalmanIMU.h.

References oCpt::components::sensors::StateKalmanIMU< T >::PhiPrime.

6.49.4.12 pos_x() [1/2]

```
template<typename T >
T oCpt::components::sensors::StateKalmanIMU< T >::pos_x ( ) const [inline]
```

Definition at line 38 of file KalmanIMU.h.

References oCpt::components::sensors::StateKalmanIMU< T >::posX.

6.49.4.13 pos_x() [2/2]

```
template<typename T >
T& oCpt::components::sensors::StateKalmanIMU< T >::pos_x ( ) [inline]
```

Definition at line 68 of file KalmanIMU.h.

References oCpt::components::sensors::StateKalmanIMU< T >::posX.

6.49.4.14 pos_y() [1/2]

```
template<typename T >
T oCpt::components::sensors::StateKalmanIMU< T >::pos_y ( ) const [inline]
```

Definition at line 40 of file KalmanIMU.h.

References oCpt::components::sensors::StateKalmanIMU< T >::posY.

6.49.4.15 pos_y() [2/2]

```
template<typename T >
T& oCpt::components::sensors::StateKalmanIMU< T >::pos_y ( ) [inline]
```

Definition at line 70 of file KalmanIMU.h.

References oCpt::components::sensors::StateKalmanIMU< T >::posY.

6.49.4.16 pos_z() [1/2]

```
template<typename T >
T oCpt::components::sensors::StateKalmanIMU< T >::pos_z ( ) const [inline]
```

Definition at line 42 of file KalmanIMU.h.

References oCpt::components::sensors::StateKalmanIMU< T >::posZ.

6.49.4.17 pos_z() [2/2]

```
template<typename T >
T& oCpt::components::sensors::StateKalmanIMU< T >::pos_z ( ) [inline]
```

Definition at line 72 of file KalmanIMU.h.

References oCpt::components::sensors::StateKalmanIMU< T >::posZ.

6.49.4.18 psi() [1/2]

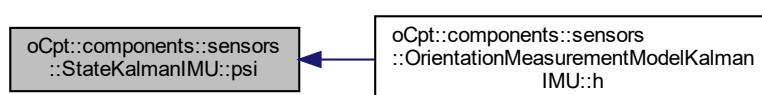
```
template<typename T >
T oCpt::components::sensors::StateKalmanIMU< T >::psi ( ) const [inline]
```

Definition at line 58 of file KalmanIMU.h.

References oCpt::components::sensors::StateKalmanIMU< T >::Psi.

Referenced by oCpt::components::sensors::OrientationMeasurementModelKalmanIMU< T, CovarianceBase >::h().

Here is the caller graph for this function:



6.49.4.19 psi() [2/2]

```
template<typename T >
T& oCpt::components::sensors::StateKalmanIMU< T >::psi ( ) [inline]
```

Definition at line 88 of file KalmanIMU.h.

References oCpt::components::sensors::StateKalmanIMU< T >::Psi.

6.49.4.20 psiPrime() [1/2]

```
template<typename T >
T oCpt::components::sensors::StateKalmanIMU< T >::psiPrime ( ) const [inline]
```

Definition at line 64 of file KalmanIMU.h.

References oCpt::components::sensors::StateKalmanIMU< T >::PsiPrime.

Referenced by oCpt::components::sensors::OrientationMeasurementModelKalmanIMU< T, CovarianceBase >↔::h().

Here is the caller graph for this function:



6.49.4.21 psiPrime() [2/2]

```
template<typename T >
T& oCpt::components::sensors::StateKalmanIMU< T >::psiPrime ( ) [inline]
```

Definition at line 94 of file KalmanIMU.h.

References oCpt::components::sensors::StateKalmanIMU< T >::PsiPrime.

6.49.4.22 theta() [1/2]

```
template<typename T >
T oCpt::components::sensors::StateKalmanIMU< T >::theta ( ) const [inline]
```

Definition at line 56 of file KalmanIMU.h.

References oCpt::components::sensors::StateKalmanIMU< T >::Theta.

Referenced by oCpt::components::sensors::OrientationMeasurementModelKalmanIMU< T, CovarianceBase >::h().

Here is the caller graph for this function:



6.49.4.23 theta() [2/2]

```
template<typename T >
T& oCpt::components::sensors::StateKalmanIMU< T >::theta ( ) [inline]
```

Definition at line 86 of file KalmanIMU.h.

References oCpt::components::sensors::StateKalmanIMU< T >::Theta.

6.49.4.24 thetaPrime() [1/2]

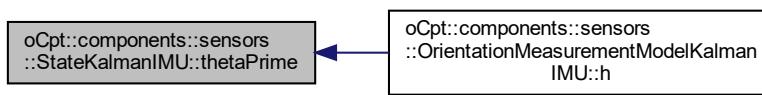
```
template<typename T >
T oCpt::components::sensors::StateKalmanIMU< T >::thetaPrime ( ) const [inline]
```

Definition at line 62 of file KalmanIMU.h.

References oCpt::components::sensors::StateKalmanIMU< T >::ThetaPrime.

Referenced by oCpt::components::sensors::OrientationMeasurementModelKalmanIMU< T, CovarianceBase >::h().

Here is the caller graph for this function:



6.49.4.25 thetaPrime() [2/2]

```
template<typename T >
T& oCpt::components::sensors::StateKalmanIMU< T >::thetaPrime ( ) [inline]
```

Definition at line 92 of file KalmanIMU.h.

References oCpt::components::sensors::StateKalmanIMU< T >::ThetaPrime.

6.49.4.26 vel_x() [1/2]

```
template<typename T >
T oCpt::components::sensors::StateKalmanIMU< T >::vel_x ( ) const [inline]
```

Definition at line 44 of file KalmanIMU.h.

References oCpt::components::sensors::StateKalmanIMU< T >::velX.

6.49.4.27 vel_x() [2/2]

```
template<typename T >
T& oCpt::components::sensors::StateKalmanIMU< T >::vel_x ( ) [inline]
```

Definition at line 74 of file KalmanIMU.h.

References oCpt::components::sensors::StateKalmanIMU< T >::velX.

6.49.4.28 vel_y() [1/2]

```
template<typename T >
T oCpt::components::sensors::StateKalmanIMU< T >::vel_y ( ) const [inline]
```

Definition at line 46 of file KalmanIMU.h.

References oCpt::components::sensors::StateKalmanIMU< T >::velY.

6.49.4.29 vel_y() [2/2]

```
template<typename T >
T& oCpt::components::sensors::StateKalmanIMU< T >::vel_y ( ) [inline]
```

Definition at line 76 of file KalmanIMU.h.

References oCpt::components::sensors::StateKalmanIMU< T >::velY.

6.49.4.30 vel_z() [1/2]

```
template<typename T >
T oCpt::components::sensors::StateKalmanIMU< T >::vel_z ( ) const [inline]
```

Definition at line 48 of file KalmanIMU.h.

References oCpt::components::sensors::StateKalmanIMU< T >::velZ.

6.49.4.31 vel_z() [2/2]

```
template<typename T >
T& oCpt::components::sensors::StateKalmanIMU< T >::vel_z ( ) [inline]
```

Definition at line 78 of file KalmanIMU.h.

References oCpt::components::sensors::StateKalmanIMU< T >::velZ.

6.49.5 Member Data Documentation

6.49.5.1 accX

```
template<typename T >
constexpr size_t oCpt::components::sensors::StateKalmanIMU< T >::accX = 6 [static]
```

Definition at line 28 of file KalmanIMU.h.

Referenced by oCpt::components::sensors::StateKalmanIMU< T >::acc_x(), and oCpt::components::sensors::PositionMeasurementKalmanIMU< T >::acc_x().

6.49.5.2 accY

```
template<typename T >
constexpr size_t oCpt::components::sensors::StateKalmanIMU< T >::accY = 7 [static]
```

Definition at line 29 of file KalmanIMU.h.

Referenced by oCpt::components::sensors::StateKalmanIMU< T >::acc_y(), and oCpt::components::sensors::PositionMeasurementKalmanIMU< T >::acc_y().

6.49.5.3 accZ

```
template<typename T >
constexpr size_t oCpt::components::sensors::StateKalmanIMU< T >::accZ = 8 [static]
```

Definition at line 30 of file KalmanIMU.h.

Referenced by oCpt::components::sensors::StateKalmanIMU< T >::acc_z(), and oCpt::components::sensors::PositionMeasurementKalmanIMU< T >::acc_z().

6.49.5.4 Phi

```
template<typename T >
constexpr size_t oCpt::components::sensors::StateKalmanIMU< T >::Phi = 11 [static]
```

Definition at line 33 of file KalmanIMU.h.

Referenced by oCpt::components::sensors::StateKalmanIMU< T >::phi(), and oCpt::components::sensors::OrientationMeasurementKalmanIMU< T >::phi().

6.49.5.5 PhiPrime

```
template<typename T >
constexpr size_t oCpt::components::sensors::StateKalmanIMU< T >::PhiPrime = 14 [static]
```

Definition at line 36 of file KalmanIMU.h.

Referenced by oCpt::components::sensors::StateKalmanIMU< T >::phiPrime(), and oCpt::components::sensors<->::OrientationMeasurementKalmanIMU< T >::phiPrime().

6.49.5.6 posX

```
template<typename T >
constexpr size_t oCpt::components::sensors::StateKalmanIMU< T >::posX = 0 [static]
```

Definition at line 22 of file KalmanIMU.h.

Referenced by oCpt::components::sensors::StateKalmanIMU< T >::pos_x().

6.49.5.7 posY

```
template<typename T >
constexpr size_t oCpt::components::sensors::StateKalmanIMU< T >::posY = 1 [static]
```

Definition at line 23 of file KalmanIMU.h.

Referenced by oCpt::components::sensors::StateKalmanIMU< T >::pos_y().

6.49.5.8 posZ

```
template<typename T >
constexpr size_t oCpt::components::sensors::StateKalmanIMU< T >::posZ = 2 [static]
```

Definition at line 24 of file KalmanIMU.h.

Referenced by oCpt::components::sensors::StateKalmanIMU< T >::pos_z().

6.49.5.9 Psi

```
template<typename T >
constexpr size_t oCpt::components::sensors::StateKalmanIMU< T >::Psi = 10 [static]
```

Definition at line 32 of file KalmanIMU.h.

Referenced by oCpt::components::sensors::StateKalmanIMU< T >::psi(), and oCpt::components::sensors<->::OrientationMeasurementKalmanIMU< T >::psi().

6.49.5.10 PsiPrime

```
template<typename T >
constexpr size_t oCpt::components::sensors::StateKalmanIMU< T >::PsiPrime = 13 [static]
```

Definition at line 35 of file KalmanIMU.h.

Referenced by oCpt::components::sensors::StateKalmanIMU< T >::psiPrime(), and oCpt::components::sensors<::OrientationMeasurementKalmanIMU< T >::psiPrime().

6.49.5.11 Theta

```
template<typename T >
constexpr size_t oCpt::components::sensors::StateKalmanIMU< T >::Theta = 9 [static]
```

Definition at line 31 of file KalmanIMU.h.

Referenced by oCpt::components::sensors::StateKalmanIMU< T >::theta(), and oCpt::components::sensors<::OrientationMeasurementKalmanIMU< T >::theta().

6.49.5.12 ThetaPrime

```
template<typename T >
constexpr size_t oCpt::components::sensors::StateKalmanIMU< T >::ThetaPrime = 12 [static]
```

Definition at line 34 of file KalmanIMU.h.

Referenced by oCpt::components::sensors::StateKalmanIMU< T >::thetaPrime(), and oCpt::components<::sensors::OrientationMeasurementKalmanIMU< T >::thetaPrime().

6.49.5.13 velX

```
template<typename T >
constexpr size_t oCpt::components::sensors::StateKalmanIMU< T >::velX = 3 [static]
```

Definition at line 25 of file KalmanIMU.h.

Referenced by oCpt::components::sensors::StateKalmanIMU< T >::vel_x().

6.49.5.14 velY

```
template<typename T >
constexpr size_t oCpt::components::sensors::StateKalmanIMU< T >::velY = 4 [static]
```

Definition at line 26 of file KalmanIMU.h.

Referenced by oCpt::components::sensors::StateKalmanIMU< T >::vel_y().

6.49.5.15 velZ

```
template<typename T >
constexpr size_t oCpt::components::sensors::StateKalmanIMU< T >::velZ = 5 [static]
```

Definition at line 27 of file KalmanIMU.h.

Referenced by oCpt::components::sensors::StateKalmanIMU< T >::vel_z().

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

- /projects/mti/ohCaptain/ohCaptain/include/Sensors/KalmanIMU.h

6.50 oCpt::iTask::Status Class Reference

```
#include <Task.h>
```

Public Types

- **typedef boost::shared_ptr< iTask::Status > ptr**
Boost shared_ptr to the task status.

Public Member Functions

- **Status ()**
- **virtual ~Status ()**
- **double progress ()**
- **bool running ()**
- **bool successful ()**

Private Attributes

- **double _progress = 0.0**
- **bool _running = false**
- **bool _successful**

6.50.1 Detailed Description

Definition at line 27 of file Task.h.

6.50.2 Member Typedef Documentation

6.50.2.1 ptr

```
typedef boost::shared_ptr<iTask::Status> oCpt::iTask::Status::ptr
```

Boost shared_ptr to the task status.

Definition at line 30 of file Task.h.

6.50.3 Constructor & Destructor Documentation

6.50.3.1 Status()

```
oCpt::iTask::Status::Status ( )
```

Constructor of the [iTask](#)

Returns

Definition at line 15 of file Task.cpp.

6.50.3.2 ~Status()

```
oCpt::iTask::Status::~Status ( ) [virtual]
```

Deconstructor

Definition at line 17 of file Task.cpp.

6.50.4 Member Function Documentation

6.50.4.1 progress()

```
double oCpt::iTask::Status::progress ( )
```

Show the progress of the task

Returns

double between 0..1

Definition at line 19 of file Task.cpp.

6.50.4.2 running()

```
bool oCpt::iTask::Status::running ( )
```

Returns the running state of the task

Returns

bool where running is true

Definition at line 21 of file Task.cpp.

6.50.4.3 successful()

```
bool oCpt::iTask::Status::successful ( )
```

Returns if the task was completed successfully

Returns

bool where a successfully completed task is true, task in progress or failed are false

Definition at line 23 of file Task.cpp.

6.50.5 Member Data Documentation

6.50.5.1 _progress

```
double oCpt::iTask::Status::_progress = 0.0 [private]
```

Definition at line 63 of file Task.h.

6.50.5.2 _running

```
bool oCpt::iTask::Status::_running = false [private]
```

Definition at line 64 of file Task.h.

6.50.5.3 _successful

```
bool oCpt::iTask::Status::_successful [private]
```

Initial value:

```
=  
false
```

Definition at line 65 of file Task.h.

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

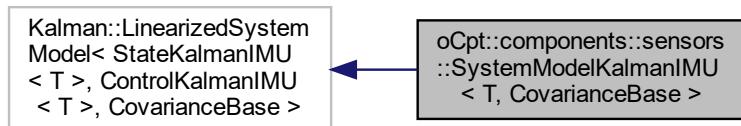
- /projects/mti/ohCaptain/ohCaptain/include/Core/[Task.h](#)
- /projects/mti/ohCaptain/ohCaptain/src/Core/[Task.cpp](#)

6.51 oCpt::components::sensors::SystemModelKalmanIMU< T, CovarianceBase >

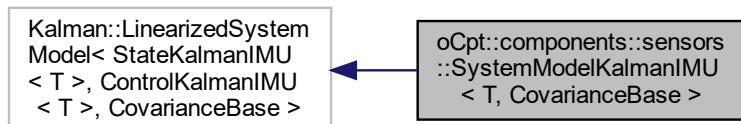
Class Template Reference

```
#include <KalmanIMU.h>
```

Inheritance diagram for oCpt::components::sensors::SystemModelKalmanIMU< T, CovarianceBase >:



Collaboration diagram for oCpt::components::sensors::SystemModelKalmanIMU< T, CovarianceBase >:



Public Types

- `typedef StateKalmanIMU< T > S`
- `typedef ControlKalmanIMU< T > C`

Public Member Functions

- `S f (const S &x, const C &u) const`

6.51.1 Detailed Description

```
template<typename T, template< class > class CovarianceBase = Kalman::StandardBase>
class oCpt::components::sensors::SystemModelKalmanIMU< T, CovarianceBase >
```

Definition at line 107 of file KalmanIMU.h.

6.51.2 Member Typedef Documentation

6.51.2.1 C

```
template<typename T , template< class > class CovarianceBase = Kalman::StandardBase>
typedef ControlKalmanIMU<T> oCpt::components::sensors::SystemModelKalmanIMU< T, CovarianceBase >::C
```

Definition at line 111 of file KalmanIMU.h.

6.51.2.2 S

```
template<typename T , template< class > class CovarianceBase = Kalman::StandardBase>
typedef StateKalmanIMU<T> oCpt::components::sensors::SystemModelKalmanIMU< T, CovarianceBase >::S
```

Definition at line 110 of file KalmanIMU.h.

6.51.3 Member Function Documentation

6.51.3.1 f()

```
template<typename T , template< class > class CovarianceBase = Kalman::StandardBase>
S oCpt::components::sensors::SystemModelKalmanIMU< T, CovarianceBase >::f (
    const S & x,
    const C & u ) const [inline]
```

Definition at line 113 of file KalmanIMU.h.

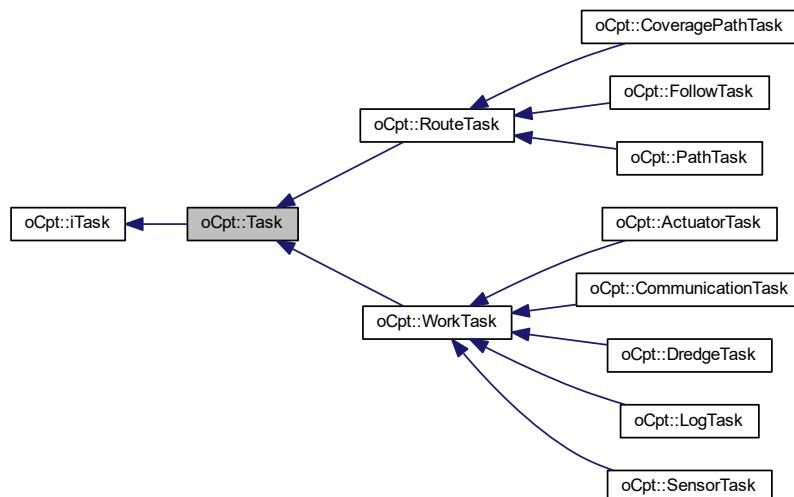
The documentation for this class was generated from the following file:

- /projects/mti/ohCaptain/ohCaptain/include/Sensors/KalmanIMU.h

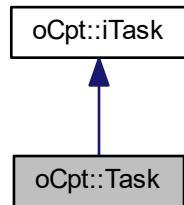
6.52 oCpt::Task Class Reference

```
#include <Task.h>
```

Inheritance diagram for oCpt::Task:



Collaboration diagram for oCpt::Task:



Public Member Functions

- `Task (Vessel::ptr vessel, bool concurrent=false)`
- `virtual ~Task ()`
- `virtual void start ()`
- `virtual iTask::Status::ptr status ()`
- `virtual void stop ()`

Protected Attributes

- `iTask::Status::ptr _status`
a boost share_ptr to the status of a task
- `TypeOf _typeof`
Indicating the type of a task.

Additional Inherited Members

6.52.1 Detailed Description

The Base [Task](#) class

Definition at line 111 of file Task.h.

6.52.2 Constructor & Destructor Documentation

6.52.2.1 Task()

```

oCpt::Task::Task (
    Vessel::ptr vessel,
    bool concurrent = false )
  
```

The constructor

Returns

Definition at line 25 of file Task.cpp.

References `_status`.

6.52.2.2 ~Task()

```
oCpt::Task::~Task ( ) [virtual]
```

The deconstructor

Definition at line 29 of file Task.cpp.

6.52.3 Member Function Documentation

6.52.3.1 start()

```
void oCpt::Task::start ( ) [virtual]
```

The start command for a task

Implements [oCpt::iTask](#).

Definition at line 31 of file Task.cpp.

References [oCpt::iTask::Work](#).

6.52.3.2 status()

```
iTask::Status::ptr oCpt::Task::status ( ) [virtual]
```

Retrieves the Status of a task

Returns

Boost shared_ptr of the task status

Implements [oCpt::iTask](#).

Definition at line 37 of file Task.cpp.

References [_status](#).

6.52.3.3 stop()

```
void oCpt::Task::stop ( ) [virtual]
```

The stop command for a task

Implements [oCpt::iTask](#).

Definition at line 39 of file Task.cpp.

6.52.4 Member Data Documentation

6.52.4.1 `_status`

`iTask::Status::ptr oCpt::Task::_status [protected]`

a boost share_ptr to the status of a task

Definition at line 141 of file Task.h.

Referenced by `status()`, and `Task()`.

6.52.4.2 `_typeof`

`TypeOf oCpt::Task::_typeof [protected]`

Indicating the type of a task.

Definition at line 142 of file Task.h.

Referenced by `oCpt::RouteTask::RouteTask()`, and `oCpt::WorkTask::WorkTask()`.

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

- /projects/mti/ohCaptain/ohCaptain/include/Core/[Task.h](#)
- /projects/mti/ohCaptain/ohCaptain/src/Core/[Task.cpp](#)

6.53 `oCpt::World::Time` Class Reference

#include <World.h>

Classes

- class [Log](#)

Public Types

- `typedef boost::shared_ptr< Time > ptr`
`Boost shared_ptr to a Time class.`
- `typedef boost::chrono::steady_clock::period tick_period`
`a tick period for a steady clock`
- `typedef boost::chrono::steady_clock clock_t`
- `typedef boost::chrono::time_point< clock_t > timepoint_t`
- `template<typename T >`
`using History = std::vector< boost::shared_ptr< Log< T >>>`

Public Member Functions

- `Time ()`
- `virtual ~Time ()`
- `clock_t & getTimeClock ()`
- `timepoint_t now ()`

Private Attributes

- `clock_t timeClock_`

6.53.1 Detailed Description

The `Time` class all things time related, which allow for easy consite time manupulation trhough out the classes
TODO add elapsed time function with operator+ and operator- to return quantity<seconds>

Definition at line 35 of file World.h.

6.53.2 Member Typedef Documentation

6.53.2.1 `clock_t`

```
typedef boost::chrono::steady_clock oCpt::World::Time::clock_t
```

Definition at line 39 of file World.h.

6.53.2.2 `History`

```
template<typename T >
using oCpt::World::Time::History = std::vector<boost::shared_ptr<Log<T>>>
```

Definition at line 105 of file World.h.

6.53.2.3 `ptr`

```
typedef boost::shared_ptr<Time> oCpt::World::Time::ptr
```

Boost shared_ptr to a `Time` class.

Definition at line 37 of file World.h.

6.53.2.4 `tick_period`

```
typedef boost::chrono::steady_clock::period oCpt::World::Time::tick_period
```

a tick period for a steady clock

Definition at line 38 of file World.h.

6.53.2.5 `timepoint_t`

```
typedef boost::chrono::time_point<clock_t> oCpt::World::Time::timepoint_t
```

Definition at line 40 of file World.h.

6.53.3 Constructor & Destructor Documentation

6.53.3.1 `Time()`

```
oCpt::World::Time::Time ( )
```

Constructor of the [Time](#) class

Definition at line 11 of file World.cpp.

6.53.3.2 `~Time()`

```
oCpt::World::Time::~Time ( ) [virtual]
```

Deconstructor of the [Time](#) class

Definition at line 13 of file World.cpp.

6.53.4 Member Function Documentation

6.53.4.1 `getTimeClock()`

```
World::Time::clock_t & oCpt::World::Time::getTimeClock ( )
```

get the current TimeClock

Returns

returns the time clock

Definition at line 15 of file World.cpp.

References `timeClock_`.

6.53.4.2 now()

```
World::Time::timepoint_t oCpt::World::Time::now ( )
```

Get the current time, as in now

Returns

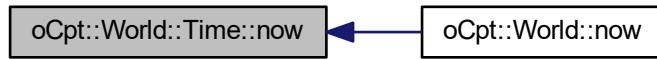
returns a timepoint_t which is now

Definition at line 19 of file World.cpp.

References timeClock_.

Referenced by oCpt::World::now().

Here is the caller graph for this function:



6.53.5 Member Data Documentation

6.53.5.1 timeClock_

```
clock_t oCpt::World::Time::timeClock_ [private]
```

Definition at line 42 of file World.h.

Referenced by getTimeClock(), and now().

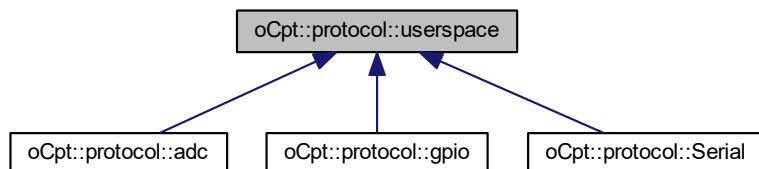
The documentation for this class was generated from the following files:

- /projects/mti/ohCaptain/ohCaptain/include/Core/World.h
- /projects/mti/ohCaptain/ohCaptain/src/Core/World.cpp

6.54 oCpt::protocol::userspace Class Reference

```
#include <Controller.h>
```

Inheritance diagram for oCpt::protocol::userspace:



Public Member Functions

- `userspace ()`
- `virtual ~userspace ()`

Protected Member Functions

- `bool modLoaded (std::string modName)`
- `bool fileExist (std::string fileName)`
- `bool dtboLoaded (std::string dtboName)`

Protected Attributes

- `std::mutex usMutex`

6.54.1 Detailed Description

Functions and routines related to the Linux userspace. Checking if a file exist, if capes or modules are loaded etc.

Definition at line 45 of file Controller.h.

6.54.2 Constructor & Destructor Documentation

6.54.2.1 `userspace()`

```
oCpt::protocol::userspace::userspace ( )
```

The constructor

Definition at line 16 of file Controller.cpp.

6.54.2.2 `~userspace()`

```
oCpt::protocol::userspace::~userspace ( ) [virtual]
```

The deconstructor

Definition at line 18 of file Controller.cpp.

6.54.3 Member Function Documentation

6.54.3.1 `dtboLoaded()`

```
bool oCpt::protocol::userspace::dtboLoaded (
    std::string dtboName ) [protected]
```

Checks if a Device Tree overlay is loaded

Parameters

<i>dtboName</i>	The devicetree overlay as a string
-----------------	------------------------------------

Returns

either true or false

Definition at line 45 of file Controller.cpp.

References BBB_CAPE_MNGR.

6.54.3.2 fileExist()

```
bool oCpt::protocol::userspace::fileExist (
    std::string fileName ) [protected]
```

Checks if a file exist

Parameters

<i>fileName</i>	the filename as string
-----------------	------------------------

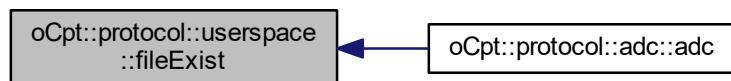
Returns

either true or false

Definition at line 39 of file Controller.cpp.

Referenced by oCpt::protocol::adc::adc().

Here is the caller graph for this function:



6.54.3.3 modLoaded()

```
bool oCpt::protocol::userspace::modLoaded (
    std::string modName ) [protected]
```

Checks if a Linux module is loaded

Parameters

<i>modName</i>	the name of the module as string
----------------	----------------------------------

Returns

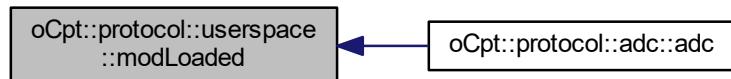
either true or false

Definition at line 20 of file Controller.cpp.

References MODULE_PATH.

Referenced by oCpt::protocol::adc::adc().

Here is the caller graph for this function:



6.54.4 Member Data Documentation

6.54.4.1 usMutex

```
std::mutex oCpt::protocol::userspace::usMutex [protected]
```

The standard Mutex TODO check if this is really needed for the current setup

Definition at line 83 of file Controller.h.

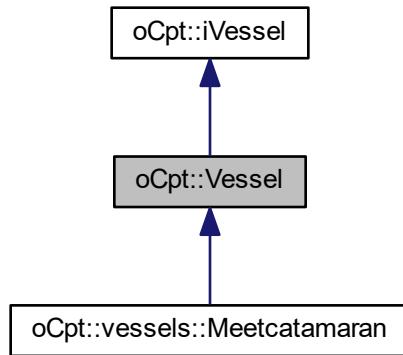
The documentation for this class was generated from the following files:

- /projects/mti/ohCaptain/ohCaptain/include/Core/Controller.h
- /projects/mti/ohCaptain/ohCaptain/src/Core/Controller.cpp

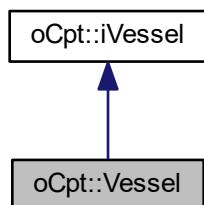
6.55 oCpt::Vessel Class Reference

```
#include <Vessel.h>
```

Inheritance diagram for oCpt::Vessel:



Collaboration diagram for oCpt::Vessel:



Public Member Functions

- [Vessel \(\)](#)
- [Vessel \(iController::ptr controller\)](#)
- virtual [~Vessel \(\)](#)
- virtual void [initialize \(\)](#) override
- virtual void [run \(\)](#) override
- virtual void [stop \(\)](#) override

Protected Attributes

- `World::ptr world_`
a shared_ptr to the world needed for time and location keeping
- `iController::ptr controller_`
a shared_ptr to the controller needed for sensors, actuators and communication
- `iCaptain::ptr captain_`
The captain for strategical planning.
- `iBoatswain::ptr boatswain_`
The boatswain, the worker synchronize operations for actuators, sensors, and communication.
- `std::vector< iSensor::ptr > sensors_`
- `std::vector< iActuator::ptr > actuators_`
- `std::vector< iComm::ptr > comm_`

Additional Inherited Members

6.55.1 Detailed Description

The vessel base class

Definition at line 80 of file Vessel.h.

6.55.2 Constructor & Destructor Documentation

6.55.2.1 `Vessel()` [1/2]

```
oCpt::Vessel::Vessel ( )
```

The constructor for a vessel

Returns

Definition at line 23 of file Vessel.cpp.

References `oCpt::iVessel::stopThread_`.

6.55.2.2 `Vessel()` [2/2]

```
oCpt::Vessel::Vessel (
    iController::ptr controller )
```

The constructor for a vessel

Parameters

<code>controller</code>	shared_ptr to the controller
-------------------------	------------------------------

Returns

Definition at line 33 of file Vessel.cpp.

References boatswain_, captain_, controller_, and world_.

6.55.2.3 ~Vessel()

```
oCpt::Vessel::~Vessel ( ) [virtual]
```

The deconstructor

Definition at line 40 of file Vessel.cpp.

6.55.3 Member Function Documentation

6.55.3.1 initialize()

```
void oCpt::Vessel::initialize ( ) [override], [virtual]
```

Initialize the vessel

Implements [oCpt::iVessel](#).

Definition at line 42 of file Vessel.cpp.

References boatswain_, and captain_.

6.55.3.2 run()

```
void oCpt::Vessel::run ( ) [override], [virtual]
```

Run the vessel normal operations

Implements [oCpt::iVessel](#).

Definition at line 47 of file Vessel.cpp.

References boatswain_, captain_, and [oCpt::iBoatswain::run\(\)](#).

Here is the call graph for this function:



6.55.3.3 `stop()`

```
void oCpt::Vessel::stop ( ) [override], [virtual]
```

Stop the vessel, everything except critical parts, which are needed to survive

Implements [oCpt::iVessel](#).

Definition at line 54 of file [Vessel.cpp](#).

References [oCpt::iVessel::stopThread_](#).

6.55.4 Member Data Documentation

6.55.4.1 `actuators_`

```
std::vector<iActuator::ptr> oCpt::Vessel::actuators_ [protected]
```

Definition at line 121 of file [Vessel.h](#).

6.55.4.2 `boatswain_`

```
iBoatswain::ptr oCpt::Vessel::boatswain_ [protected]
```

The boatswain, the worker synchronize operations for actuators, sensors, and communication.

Definition at line 119 of file [Vessel.h](#).

Referenced by [initialize\(\)](#), [run\(\)](#), and [Vessel\(\)](#).

6.55.4.3 `captain_`

```
iCaptain::ptr oCpt::Vessel::captain_ [protected]
```

The captain for strategical planning.

Definition at line 118 of file [Vessel.h](#).

Referenced by [initialize\(\)](#), [run\(\)](#), and [Vessel\(\)](#).

6.55.4.4 `comm_`

```
std::vector<iComm::ptr> oCpt::Vessel::comm_ [protected]
```

Definition at line 122 of file [Vessel.h](#).

6.55.4.5 controller_

`iController::ptr oCpt::Vessel::controller_ [protected]`

a shared_ptr to the controller needed for sensors, actuators and coommunication

Definition at line 117 of file Vessel.h.

Referenced by `Vessel()`.

6.55.4.6 sensors_

`std::vector<iSensor::ptr> oCpt::Vessel::sensors_ [protected]`

Definition at line 120 of file Vessel.h.

6.55.4.7 world_

`World::ptr oCpt::Vessel::world_ [protected]`

a shared_ptr to the world needed for time and location keeping

Definition at line 116 of file Vessel.h.

Referenced by `Vessel()`.

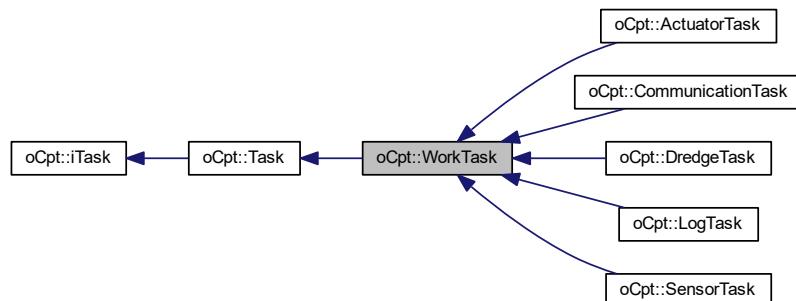
The documentation for this class was generated from the following files:

- /projects/mti/ohCaptain/ohCaptain/include/Core/[Vessel.h](#)
- /projects/mti/ohCaptain/ohCaptain/src/Core/[Vessel.cpp](#)

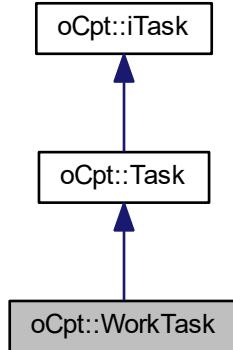
6.56 oCpt::WorkTask Class Reference

```
#include <Task.h>
```

Inheritance diagram for oCpt::WorkTask:



Collaboration diagram for oCpt::WorkTask:



Public Member Functions

- `WorkTask (Vessel::ptr vessel, bool concurrent=false)`
- virtual `~WorkTask \(\)`

Additional Inherited Members

6.56.1 Detailed Description

An object representing work related tasks

Definition at line 167 of file Task.h.

6.56.2 Constructor & Destructor Documentation

6.56.2.1 WorkTask()

```

oCpt::WorkTask::WorkTask (
    Vessel::ptr vessel,
    bool concurrent = false )
  
```

Constructor of the interface

Returns

Definition at line 47 of file Task.cpp.

References oCpt::Task::_typeof.

6.56.2.2 ~WorkTask()

```
oCpt::WorkTask::~WorkTask ( ) [virtual]
```

The deconstructor

Definition at line 51 of file Task.cpp.

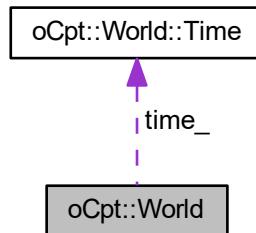
The documentation for this class was generated from the following files:

- /projects/mti/ohCaptain/ohCaptain/include/Core/[Task.h](#)
- /projects/mti/ohCaptain/ohCaptain/src/Core/[Task.cpp](#)

6.57 oCpt::World Class Reference

```
#include <World.h>
```

Collaboration diagram for oCpt::World:



Classes

- class [Location](#)
- class [Time](#)

Public Types

- `typedef boost::shared_ptr< World > ptr`
Boost shared_ptr to a [World](#) class.

Public Member Functions

- [World \(\)](#)
- virtual [~World \(\)](#)
- [Time & getTime \(\)](#)
- [Time::timepoint_t now \(\)](#)

Protected Attributes

- [Time time_](#)

6.57.1 Detailed Description

The [World](#) class, this class is an shared pointer where the boatswain can place the state representation of the vessel at a certain time, which allows the captain to plan the strategic decisions

Definition at line 28 of file World.h.

6.57.2 Member Typedef Documentation

6.57.2.1 ptr

```
typedef boost::shared_ptr<World> oCpt::World::ptr
```

Boost shared_ptr to a [World](#) class.

Definition at line 30 of file World.h.

6.57.3 Constructor & Destructor Documentation

6.57.3.1 World()

```
oCpt::World::World ( )
```

Constructor for a [World](#)

Definition at line 23 of file World.cpp.

6.57.3.2 ~World()

```
oCpt::World::~World ( ) [virtual]
```

Deconstructor for a [World](#)

Definition at line 27 of file World.cpp.

6.57.4 Member Function Documentation

6.57.4.1 getTime()

```
World::Time & oCpt::World::getTime ( )
```

get the current time object

Returns

returns [Time](#)

Definition at line 29 of file World.cpp.

References [time_](#).

6.57.4.2 now()

```
World::Time::timepoint_t oCpt::World::now ( )
```

Get the current Epoch

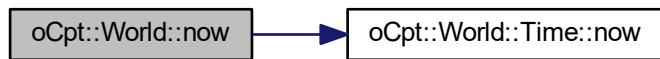
Returns

returns a timepoint representing now

Definition at line 33 of file World.cpp.

References oCpt::World::Time::now(), and time_.

Here is the call graph for this function:



6.57.5 Member Data Documentation

6.57.5.1 time_

```
Time oCpt::World::time_ [protected]
```

Definition at line 224 of file World.h.

Referenced by getTime(), and now().

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

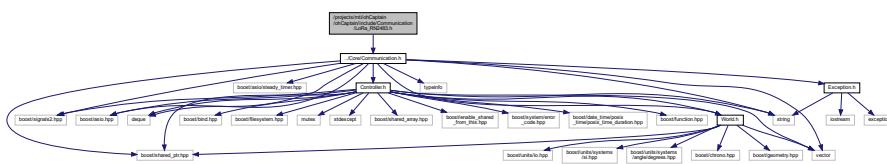
- /projects/mti/ohCaptain/ohCaptain/include/Core/[World.h](#)
- /projects/mti/ohCaptain/ohCaptain/src/Core/[World.cpp](#)

Chapter 7

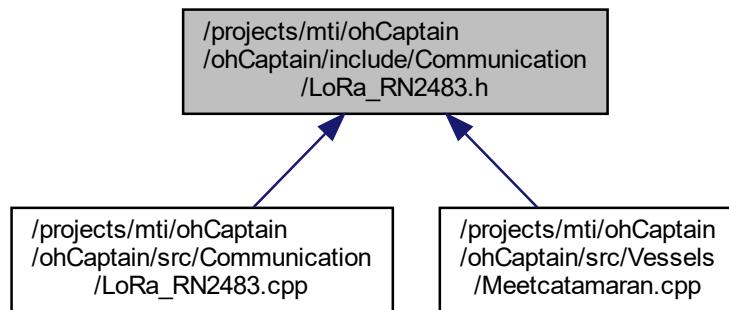
File Documentation

7.1 /projects/mti/ohCaptain/ohCaptain/include/Communication/LoRa_RN2483.h File Reference

```
#include "../Core/Communication.h"
Include dependency graph for LoRa_RN2483.h:
```



This graph shows which files directly or indirectly include this file:



Classes

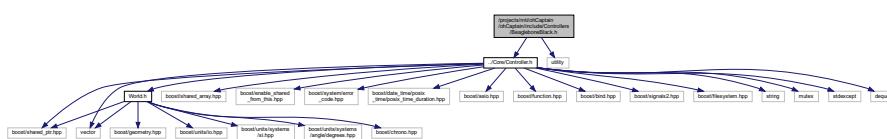
- class oCpt::components::comm::LoRa_RN2483

Namespaces

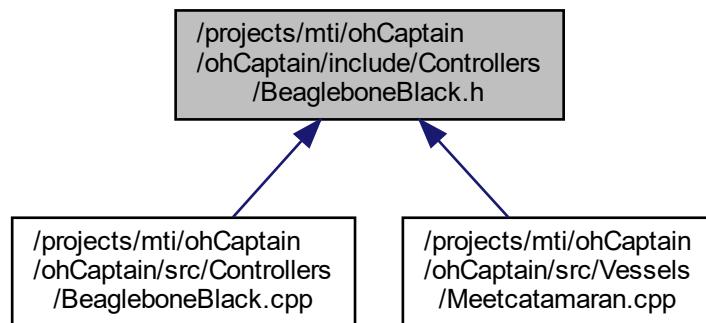
- [oCpt](#)
- [oCpt::components](#)
- [oCpt::components::comm](#)

7.2 /projects/mti/ohCaptain/ohCaptain/include/Controllers/BeagleboneBlack.h File Reference

```
#include "../Core/Controller.h"
#include <utility>
Include dependency graph for BeagleboneBlack.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- class [oCpt::components::controller::BBB](#)

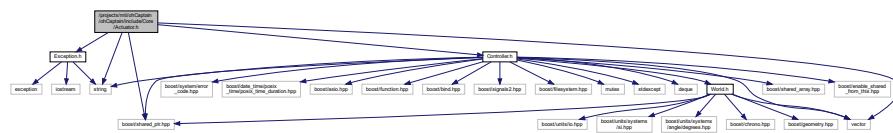
Namespaces

- [oCpt](#)
- [oCpt::components](#)
- [oCpt::components::controller](#)

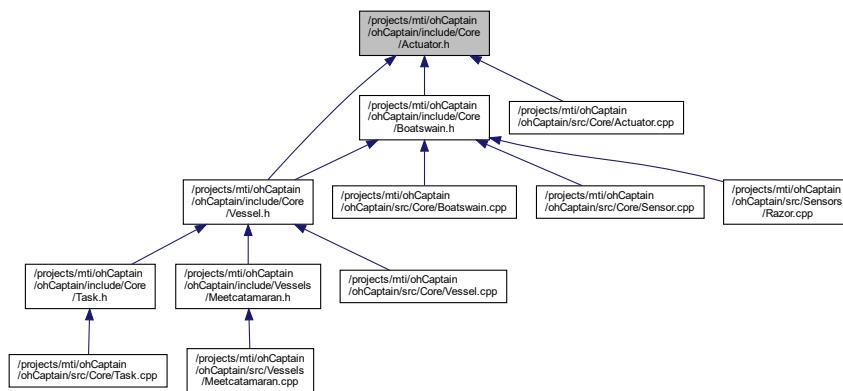
7.3 /projects/mti/ohCaptain/ohCaptain/include/Core/Actuator.h File Reference

```
#include <boost/shared_ptr.hpp>
#include <string>
#include <vector>
#include "Controller.h"
#include "Exception.h"
```

Include dependency graph for Actuator.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [oCpt::iActuator](#)
- class [oCpt::Actuator](#)

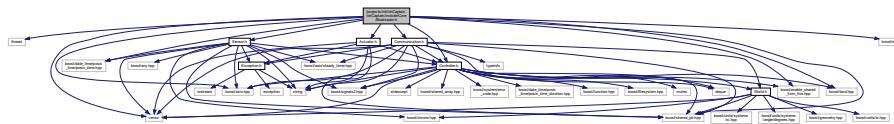
Namespaces

- [oCpt](#)

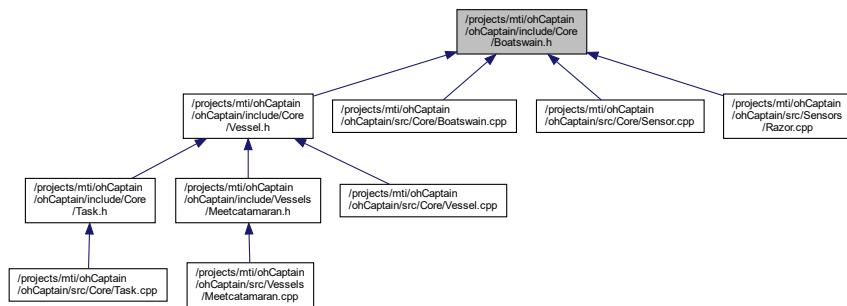
7.4 /projects/mti/ohCaptain/ohCaptain/include/Core/Boatswain.h File Reference

```
#include <thread>
#include <vector>
#include <boost/asio.hpp>
#include <boost/date_time/posix_time/posix_time.hpp>
#include <boost/shared_ptr.hpp>
```

```
#include <boost/enable_shared_from_this.hpp>
#include <boost/bind.hpp>
#include <boost/ref.hpp>
#include "Controller.h"
#include "Sensor.h"
#include "Actuator.h"
#include "Communication.h"
Include dependency graph for Boatswain.h:
```



This graph shows which files directly or indirectly include this file:



Classes

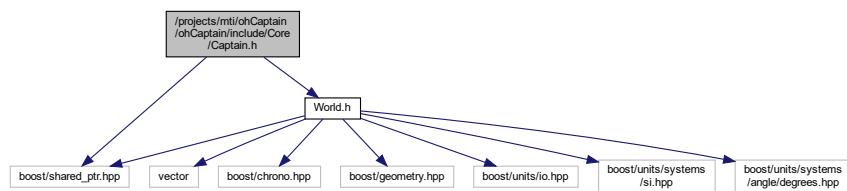
- class [oCpt::iBoatswain](#)
- class [oCpt::Boatswain](#)

Namespaces

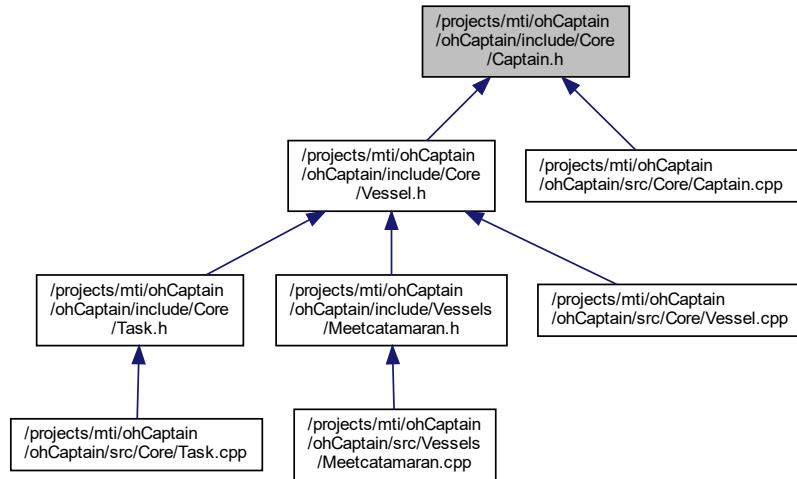
- [oCpt](#)

7.5 /projects/mti/ohCaptain/ohCaptain/include/Core/Captain.h File Reference

```
#include <boost/shared_ptr.hpp>
#include "World.h"
Include dependency graph for Captain.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- class `oCpt::iCaptain`
- class `oCpt::Captain`

Namespaces

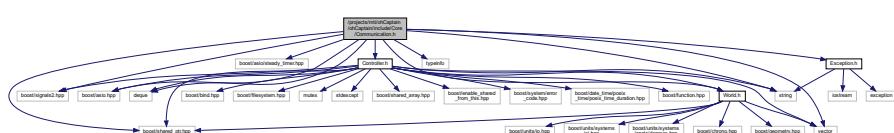
- `oCpt`

7.6 /projects/mti/ohCaptain/ohCaptain/include/Core/Communication.h File Reference

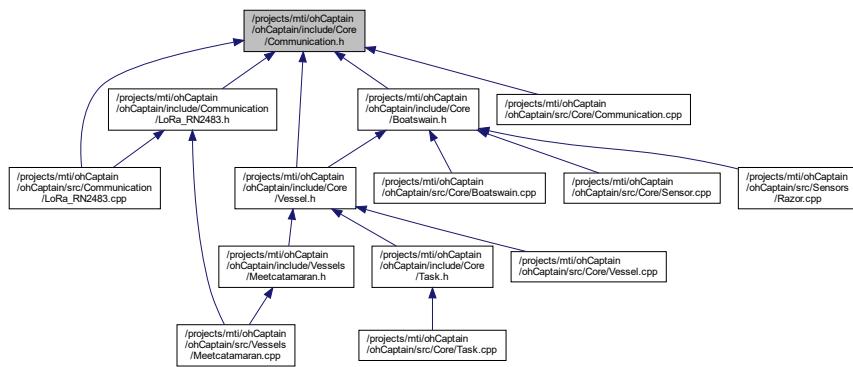
```

#include <boost/shared_ptr.hpp>
#include <boost/signals2.hpp>
#include <boost/asio/steady_timer.hpp>
#include <boost/asio.hpp>
#include <string>
#include <vector>
#include <deque>
#include <typeinfo>
#include "Controller.h"
#include "World.h"
#include "Exception.h"
  
```

Include dependency graph for `Communication.h`:



This graph shows which files directly or indirectly include this file:



Classes

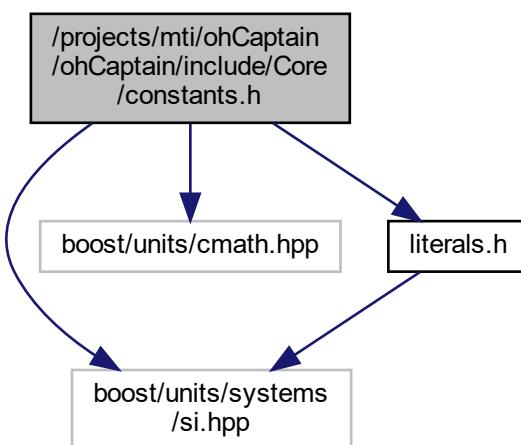
- class [oCpt::iComm](#)
- struct [oCpt::iComm::Message](#)
- class [oCpt::LoRa](#)

Namespaces

- [oCpt](#)

7.7 /projects/mti/ohCaptain/ohCaptain/include/Core/constants.h File Reference

```
#include <boost/units/systems/si.hpp>
#include <boost/units/cmath.hpp>
#include "literals.h"
Include dependency graph for constants.h:
```



Namespaces

- boost
- boost::units
- boost::units::constants

Functions

- long double boost::units::constants::operator""_LD (unsigned long long x)
- long double boost::units::constants::operator""_LD (long double x)

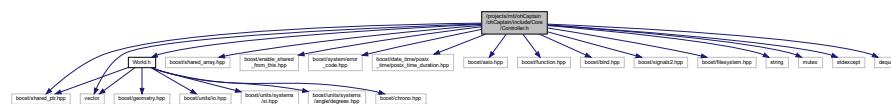
Variables

- const auto boost::units::constants::M_PI_LD = static_cast<long double>(M_PI)
- const auto boost::units::constants::c = 299792458.0_m / 1_s
- const auto boost::units::constants::G = 6.67384E-11_m * 1_m * 1_m / 1_kg / 1_s / 1_s
- const auto boost::units::constants::h = 6.62606957E-34_J * 1_s
- const auto boost::units::constants::hbar = h / 2_LD * M_PI_LD
- const auto boost::units::constants::u0 = 4E-7_N * M_PI_LD / 1_A / 1_A
- const auto boost::units::constants::eps0 = 1_LD / u0 / c / c
- const auto boost::units::constants::Z0 = u0 * c
- const auto boost::units::constants::ke = 1_LD / (4_LD * M_PI * eps0)
- const auto boost::units::constants::e = 1.602176565E-19_C
- const auto boost::units::constants::me = 9.10938291E-31_kg
- const auto boost::units::constants::mp = 1.672621777E-27_kg
- const auto boost::units::constants::uB = e * hbar / (2_LD * me)
- const auto boost::units::constants::G0 = 2_LD * e * e / h
- const auto boost::units::constants::KJ = 2_LD * e / h
- const auto boost::units::constants::uN = e * h / (2_LD * mp)
- const auto boost::units::constants::RK = h / (e * e)
- const auto boost::units::constants::alpha = u0 * e * e * c / (2_LD * h)
- const auto boost::units::constants::Rinf = alpha * alpha * me * c / (2_LD * h)
- const auto boost::units::constants::a0 = alpha / (4_LD * M_PI_LD * Rinf)
- const auto boost::units::constants::re = e * e / (4_LD * M_PI_LD * eps0 * me * c * c)
- const auto boost::units::constants::Eh = 2_LD * Rinf * h * c
- const auto boost::units::constants::R = 8.3144621_J / 1_K / 1_mol
- const auto boost::units::constants::atm = 101325_Pa
- const auto boost::units::constants::IP = sqrt(hbar * G / (c * c * c))
- const auto boost::units::constants::mP = sqrt(hbar * c / G)
- const auto boost::units::constants::tP = sqrt(hbar * G / (c * c * c * c * c))
- const auto boost::units::constants::NA = 6.02214129E23_LD / 1.0_mol
- const auto boost::units::constants::k = R / NA
- const auto boost::units::constants::kB = k
- const auto boost::units::constants::F = NA * e
- const auto boost::units::constants::c1 = 2_LD * M_PI * h * c * c
- const auto boost::units::constants::c2 = h * c / k
- const auto boost::units::constants::sigma = M_PI_LD * M_PI_LD * k * k * k * k / (60_LD * hbar * hbar * hbar * c * c)
- const auto boost::units::constants::b = h * c / (4.965114231_LD * k)
- const auto boost::units::constants::g = 9.80665_m / 1_s / 1_s

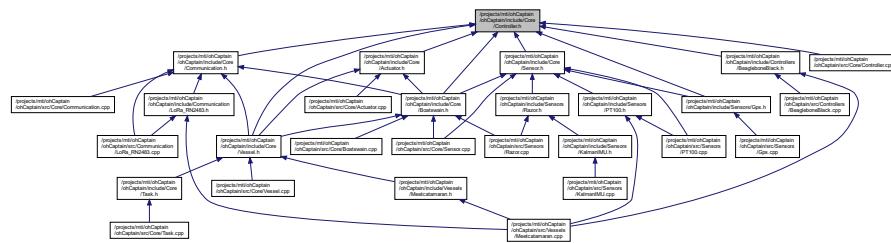
7.8 /projects/mti/ohCaptain/ohCaptain/include/Core/Controller.h File Reference

```
#include <boost/shared_ptr.hpp>
#include <boost/shared_array.hpp>
#include <boost/enable_shared_from_this.hpp>
#include <boost/system/error_code.hpp>
#include <boost/date_time posix_time posix_time_duration.hpp>
#include <boost/asio.hpp>
#include <boost/function.hpp>
#include <boost/bind.hpp>
#include <boost/signals2.hpp>
#include <boost/filesystem.hpp>
#include <string>
#include <vector>
#include <mutex>
#include <stdexcept>
#include <deque>
#include "World.h"
#include "World.h"
```

Include dependency graph for Controller.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [oCpt::protocol::userspace](#)
- class [oCpt::protocol::adc](#)
- class [oCpt::protocol::gpio](#)
- class [oCpt::protocol::Serial](#)
- class [oCpt::iController](#)
- class [oCpt::ARM](#)

Namespaces

- [oCpt](#)
- [oCpt::protocol](#)

Macros

- `#define MAX_READ_LENGTH 4096`
- `#define BBB_CAPE_MNGR "/sys/devices/platform/bone_capemgr/slots"`
- `#define GPIO_BASE_PATH "/sys/class/gpio/"`
- `#define ADC_IO_BASE_PATH "/sys/bus/iio/devices/iio:device"`
- `#define ADC_VOLTAGE_PATH "/in_voltage"`
- `#define ADC_VOLTAGE_SUB_PATH "_raw"`
- `#define MODULE_PATH "/proc/modules"`

7.8.1 Macro Definition Documentation

7.8.1.1 ADC_IO_BASE_PATH

```
#define ADC_IO_BASE_PATH "/sys/bus/iio/devices/iio:device"
```

Definition at line 32 of file Controller.h.

Referenced by oCpt::protocol::adc::adc().

7.8.1.2 ADC_VOLTAGE_PATH

```
#define ADC_VOLTAGE_PATH "/in_voltage"
```

Definition at line 33 of file Controller.h.

Referenced by oCpt::protocol::adc::adc().

7.8.1.3 ADC_VOLTAGE_SUB_PATH

```
#define ADC_VOLTAGE_SUB_PATH "_raw"
```

Definition at line 34 of file Controller.h.

Referenced by oCpt::protocol::adc::adc().

7.8.1.4 BBB_CAPE_MNGR

```
#define BBB_CAPE_MNGR "/sys/devices/platform/bone_capemgr/slots"
```

Definition at line 28 of file Controller.h.

Referenced by oCpt::protocol::userspace::dtboLoaded().

7.8.1.5 GPIO_BASE_PATH

```
#define GPIO_BASE_PATH "/sys/class/gpio/"
```

Definition at line 30 of file Controller.h.

Referenced by oCpt::protocol::gpio::exportedGpios(), oCpt::protocol::gpio::exportPin(), oCpt::protocol::gpio::gpio(), oCpt::protocol::gpio::readPinValue(), oCpt::protocol::gpio::unexportPin(), and oCpt::protocol::gpio::writePinValue().

7.8.1.6 MAX_READ_LENGTH

```
#define MAX_READ_LENGTH 4096
```

Definition at line 26 of file Controller.h.

7.8.1.7 MODULE_PATH

```
#define MODULE_PATH "/proc/modules"
```

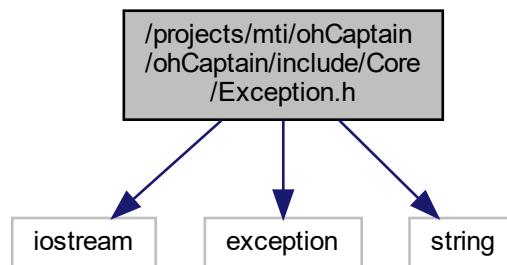
Definition at line 36 of file Controller.h.

Referenced by oCpt::protocol::userspace::modLoaded().

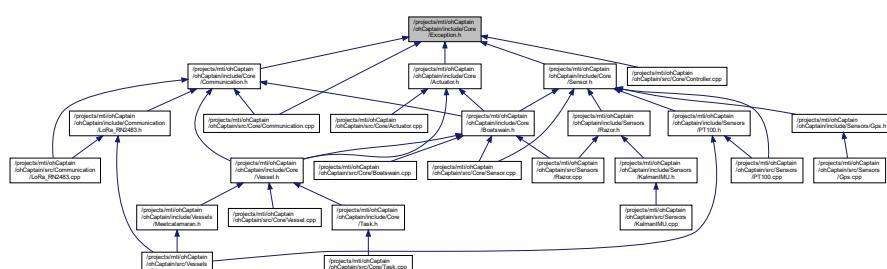
7.9 /projects/mti/ohCaptain/ohCaptain/include/Core/Exception.h File Reference

```
#include <iostream>
#include <exception>
#include <string>
```

Include dependency graph for Exception.h:



This graph shows which files directly or indirectly include this file:



Classes

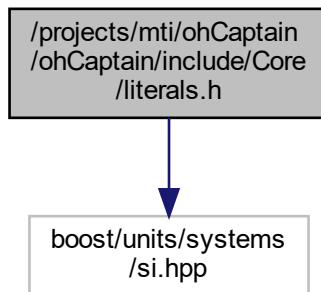
- class [oCpt::oCptException](#)

Namespaces

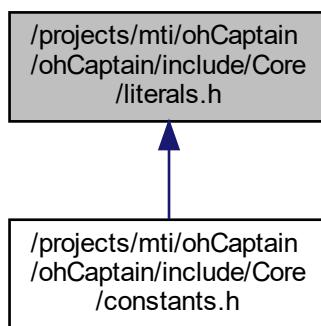
- [oCpt](#)

7.10 /projects/mti/ohCaptain/ohCaptain/include/Core/literals.h File Reference

```
#include <boost/units/systems/si.hpp>
Include dependency graph for literals.h:
```



This graph shows which files directly or indirectly include this file:



Namespaces

- boost
 - boost::units
 - boost::units::literals

Macros

- `#define BOOST_UNITS_LITERAL(suffix, unit, val, prefix, multiplier)`
 - `#define BOOST_UNITS_LITERAL_SET(suffix, unit, val)`

Functions

7.10.1 Macro Definition Documentation

7.10.1.1 BOOST_UNITS_LITERAL

```
#define BOOST_UNITS_LITERAL(
    suffix,
    unit,
    val,
    prefix,
    multiplier )
```

Value:

```
quantity<unit, long double> operator "" _##prefix##suffix(long double x) \
{ \
    return quantity<unit, long double>(x * multiplier * val); \
} \
quantity<unit, unsigned long long> operator "" _##prefix##suffix(unsigned long long x) \
{ \
    return quantity<unit, unsigned long long>(x * multiplier * val); \
}
```

Definition at line 32 of file literals.h.

7.10.1.2 BOOST_UNITS_LITERAL_SET

```
#define BOOST_UNITS_LITERAL_SET(
    suffix,
    unit,
    val )
```

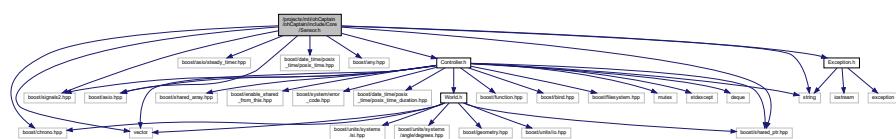
Value:

```
BOOST_UNITS_LITERAL(suffix, unit, val, Y, 1000000000000000000000000.0) \
BOOST_UNITS_LITERAL(suffix, unit, val, Z, 1000000000000000000000000.0) \
BOOST_UNITS_LITERAL(suffix, unit, val, E, 1000000000000000000000000.0) \
BOOST_UNITS_LITERAL(suffix, unit, val, P, 10000000000000000000.0) \
BOOST_UNITS_LITERAL(suffix, unit, val, T, 1000000000000.0) \
BOOST_UNITS_LITERAL(suffix, unit, val, G, 1000000000.0) \
BOOST_UNITS_LITERAL(suffix, unit, val, M, 1000000.0) \
BOOST_UNITS_LITERAL(suffix, unit, val, k, 1000.0) \
BOOST_UNITS_LITERAL(suffix, unit, val, h, 100.0) \
BOOST_UNITS_LITERAL(suffix, unit, val, da, 10.0) \
BOOST_UNITS_LITERAL(suffix, unit, val, , 1.0) \
BOOST_UNITS_LITERAL(suffix, unit, val, d, 0.1) \
BOOST_UNITS_LITERAL(suffix, unit, val, c, 0.01) \
BOOST_UNITS_LITERAL(suffix, unit, val, m, 0.001) \
BOOST_UNITS_LITERAL(suffix, unit, val, u, 0.00001) \
BOOST_UNITS_LITERAL(suffix, unit, val, n, 0.0000001) \
BOOST_UNITS_LITERAL(suffix, unit, val, p, 0.0000000001) \
BOOST_UNITS_LITERAL(suffix, unit, val, f, 0.000000000001) \
BOOST_UNITS_LITERAL(suffix, unit, val, a, 0.00000000000001) \
BOOST_UNITS_LITERAL(suffix, unit, val, z, 0.0000000000000001) \
BOOST_UNITS_LITERAL(suffix, unit, val, y, 0.0000000000000001)
```

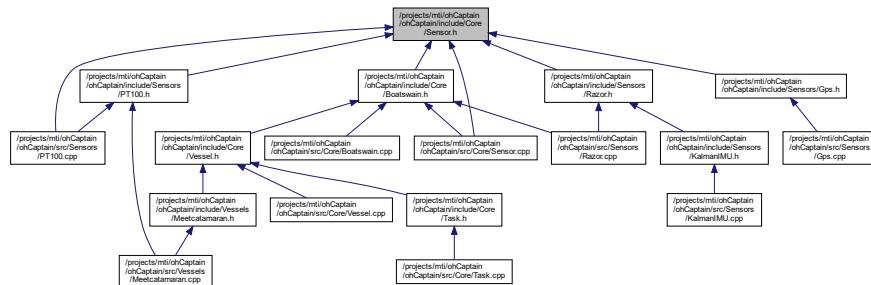
Definition at line 42 of file literals.h.

7.11 /projects/mti/ohCaptain/ohCaptain/include/Core/Sensor.h File Reference

```
#include <boost/shared_ptr.hpp>
#include <boost/signals2.hpp>
#include <boost/chrono.hpp>
#include <boost/asio/steady_timer.hpp>
#include <boost/asio.hpp>
#include <boost/date_time posix_time posix_time.hpp>
#include <boost/any.hpp>
#include <string>
#include <vector>
#include "Controller.h"
#include "Exception.h"
Include dependency graph for Sensor.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- class oCpt::iSensor
 - struct oCpt::iSensor::State
 - class oCpt::Sensor

Namespaces

- oCpt

Macros

- #define CAST(x, t)

7.11.1 Macro Definition Documentation

7.11.1.1 CAST

```
#define CAST(
    x,
    t )
```

Value:

```
boost::any_cast<t::ReturnValue_t>(x) /*<! CAST the return value of a generic boost::any object, which can
change for each sensor to a the proper return value. where the first parameter is the getState().Value and
the second is the Sensor Class.
*/

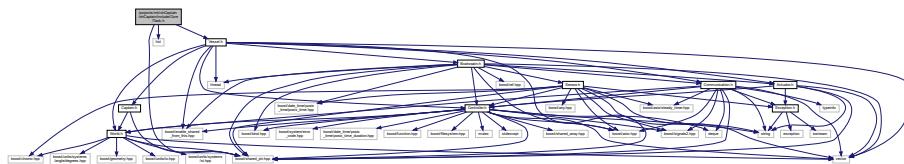
```

Definition at line 21 of file Sensor.h.

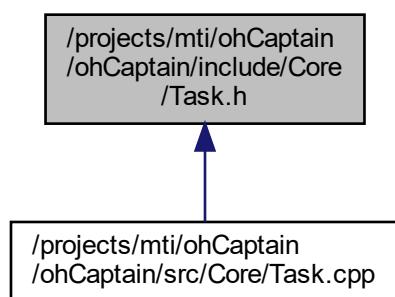
Referenced by oCpt::components::sensors::KalmanIMU::RazorUpdate().

7.12 /projects/mti/ohCaptain/ohCaptain/include/Core/Task.h File Reference

```
#include <boost/shared_ptr.hpp>
#include <list>
#include "Vessel.h"
Include dependency graph for Task.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- class [oCpt::iTask](#)
Task interface, all tasks need to adhere to this structure.
- class [oCpt::iTask::Status](#)
- class [oCpt::Task](#)
- class [oCpt::RouteTask](#)
- class [oCpt::WorkTask](#)
- class [oCpt::CoveragePathTask](#)
An object representing a coverage path task.
- class [oCpt::FollowTask](#)
An object representing a follow the target task.
- class [oCpt::PathTask](#)
An object representing a normal A to B type of path planning.
- class [oCpt::LogTask](#)
An Object representing a data logging task.
- class [oCpt::DredgeTask](#)
An Object representing a dredging task.
- class [oCpt::SensorTask](#)
- class [oCpt::ActuatorTask](#)
- class [oCpt::CommunicationTask](#)

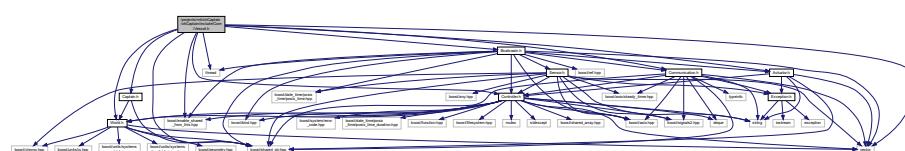
Namespaces

- [oCpt](#)

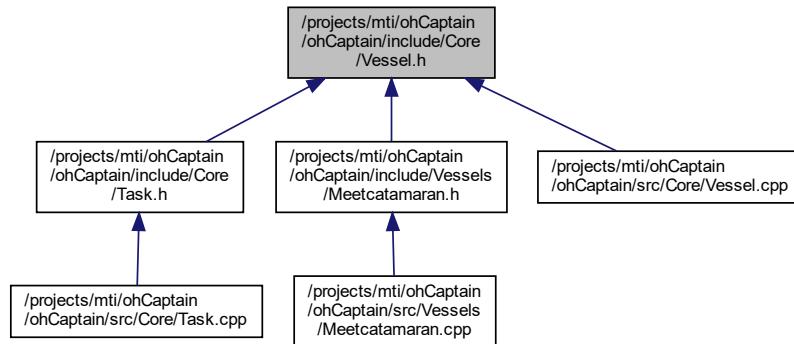
7.13 /projects/mti/ohCaptain/ohCaptain/include/Core/Vessel.h File Reference

```
#include <boost/shared_ptr.hpp>
#include <boost/enable_shared_from_this.hpp>
#include <vector>
#include <thread>
#include "World.h"
#include "Controller.h"
#include "Captain.h"
#include "Boatswain.h"
#include "Actuator.h"
#include "Communication.h"
```

Include dependency graph for Vessel.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [oCpt::iVessel](#)
- class [oCpt::Vessel](#)

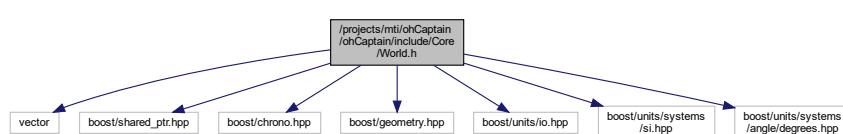
Namespaces

- [oCpt](#)

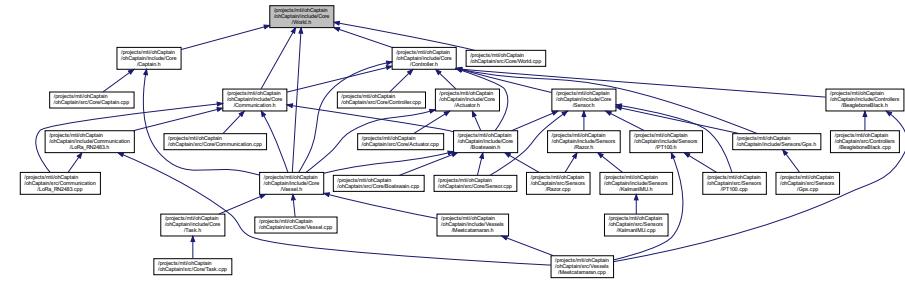
7.14 /projects/mti/ohCaptain/ohCaptain/include/Core/World.h File Reference

```

#include <vector>
#include <boost/shared_ptr.hpp>
#include <boost/chrono.hpp>
#include <boost/geometry.hpp>
#include <boost/units/io.hpp>
#include <boost/units/systems/si.hpp>
#include <boost/units/systems/angle/degrees.hpp>
Include dependency graph for World.h:
  
```



This graph shows which files directly or indirectly include this file:



Classes

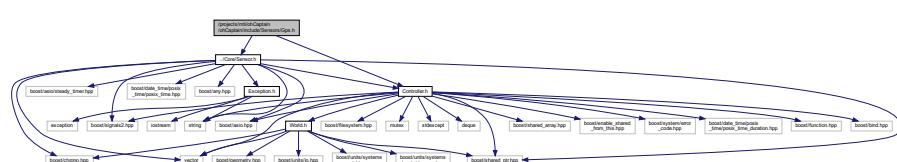
- class `oCpt::World`
 - class `oCpt::World::Time`
 - class `oCpt::World::Time::Log< T >`
 - class `oCpt::World::Location`
 - struct `oCpt::World::Location::coordinate`
 - struct `oCpt::World::Location::gpsPoint`
 - struct `oCpt::World::Location::RoutePoint`

Namespaces

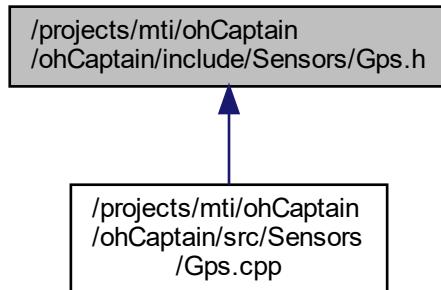
- oCpt

7.15 /projects/mti/ohCaptain/ohCaptain/include/Sensors/Gps.h File Reference

```
#include "../Core/Sensor.h"
#include "../Core/Controller.h"
Include dependency graph for Gps.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- class [oCpt::components::sensors::Gps](#)

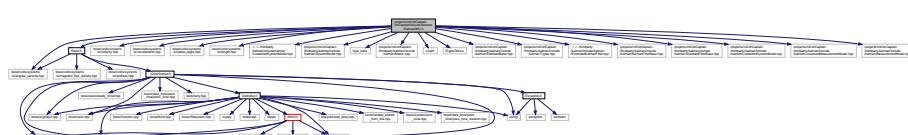
Namespaces

- [oCpt](#)
- [oCpt::components](#)
- [oCpt::components::sensors](#)

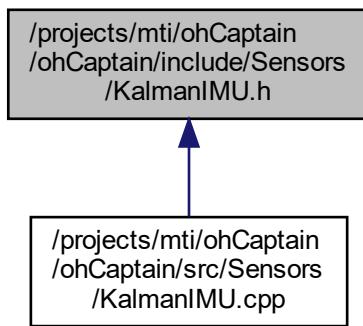
7.16 /projects/mti/ohCaptain/ohCaptain/include/Sensors/KalmanIMU.h File Reference

```

#include "Razor.h"
#include <boost/units/systems/si/velocity.hpp>
#include <boost/units/systems/si/acceleration.hpp>
#include <boost/units/systems/si/angle.hpp>
#include <boost/units/systems/si/angular_velocity.hpp>
#include <boost/units/systems/si/length.hpp>
#include "../../thirdparty/kalman/include/kalman/LinearizedSystemModel.hpp"
#include "../../thirdparty/kalman/include/kalman/ExtendedKalmanFilter.hpp"
Include dependency graph for KalmanIMU.h:
  
```



This graph shows which files directly or indirectly include this file:



Classes

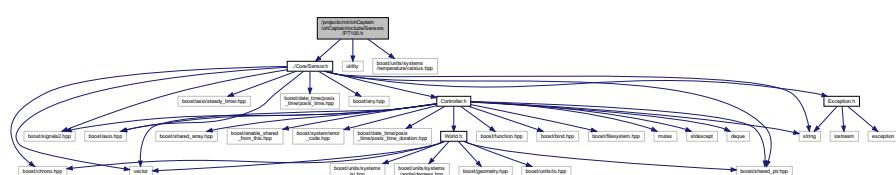
- class `oCpt::components::sensors::StateKalmanIMU< T >`
 - class `oCpt::components::sensors::ControlKalmanIMU< T >`
 - class `oCpt::components::sensors::SystemModelKalmanIMU< T, CovarianceBase >`
 - class `oCpt::components::sensors::OrientationMeasurementKalmanIMU< T >`
 - class `oCpt::components::sensors::OrientationMeasurementModelKalmanIMU< T, CovarianceBase >`
 - class `oCpt::components::sensors::PositionMeasurementKalmanIMU< T >`
 - class `oCpt::components::sensors::PositionMeasurementModelKalmanIMU< T, CovarianceBase >`
 - class `oCpt::components::sensors::KalmanIMU`
 - struct `oCpt::components::sensors::KalmanIMU::ReturnValue`

Namespaces

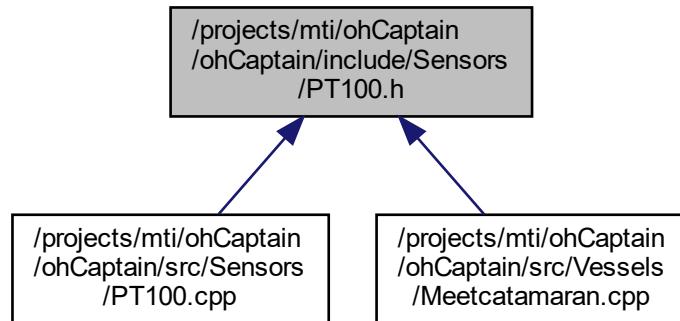
- `oCpt`
 - `oCpt::components`
 - `oCpt::components::sensors`

7.17 /projects/mti/ohCaptain/ohCaptain/include/Sensors/PT100.h File Reference

```
#include "../Core/Sensor.h"
#include <utility>
#include <boost/units/systems/temperature/celsius.hpp>
Include dependency graph for PT100.h:
```



This graph shows which files directly or indirectly include this file:



Classes

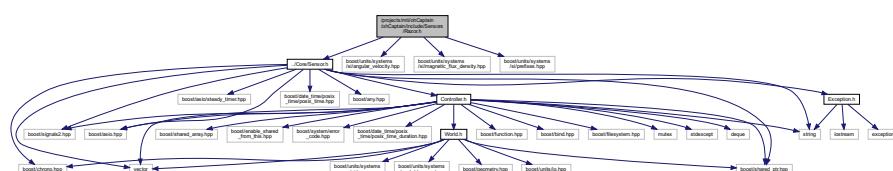
- class oCpt::components::sensors::PT100

Namespaces

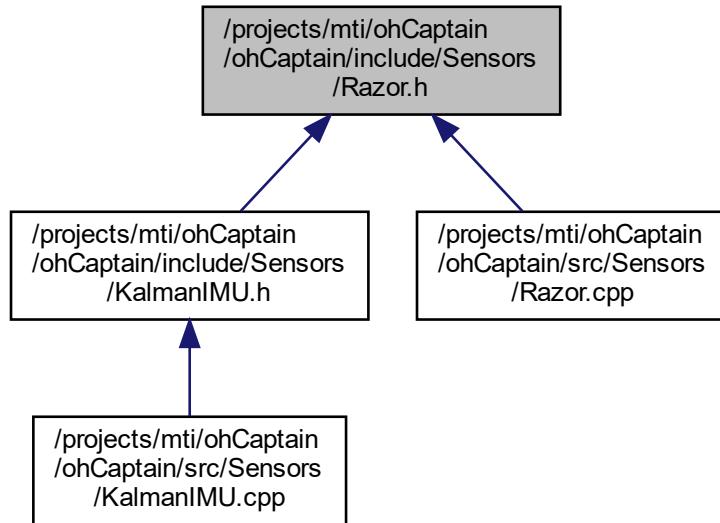
- oCpt
 - oCpt::components
 - oCpt::components::sensors

7.18 /projects/mti/ohCaptain/ohCaptain/include/Sensors/Razor.h File Reference

```
#include "../Core/Sensor.h"
#include <boost/units/systems/si/angular_velocity.hpp>
#include <boost/units/systems/si/magnetic_flux_density.hpp>
#include <boost/units/systems/si/prefixes.hpp>
Include dependency graph for Razor.h:
```



This graph shows which files directly or indirectly include this file:



Classes

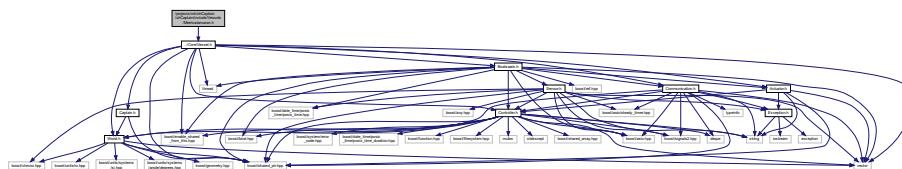
- class `oCpt::components::sensors::Razor`
- struct `oCpt::components::sensors::Razor::ReturnValue`

Namespaces

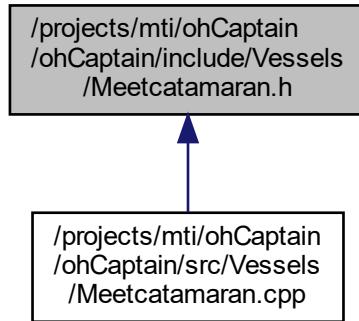
- `oCpt`
- `oCpt::components`
- `oCpt::components::sensors`

7.19 /projects/mti/ohCaptain/ohCaptain/include/Vessels/Meetcatamaran.h File Reference

```
#include "../Core/Vessel.h"
Include dependency graph for Meetcatamaran.h:
```



This graph shows which files directly or indirectly include this file:



Classes

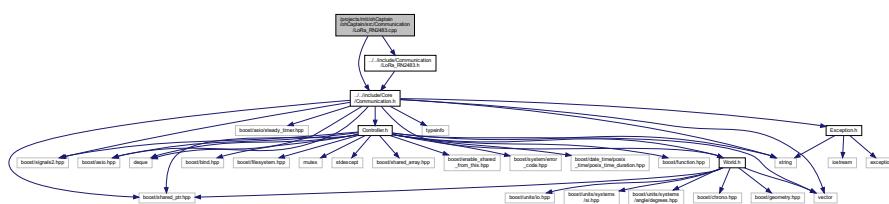
- class `oCpt::vessels::Meetcatamaran`

Namespaces

- oCpt
 - oCpt::vessels

7.20 /projects/mti/ohCaptain/ohCaptain/src/Communication/LoRa_RN2483.cpp File Reference

```
#include "../../include/Core/Communication.h"
#include "../../include/Communication/LoRa_RN2483.h"
Include dependency graph for LoRa_RN2483.cpp:
```

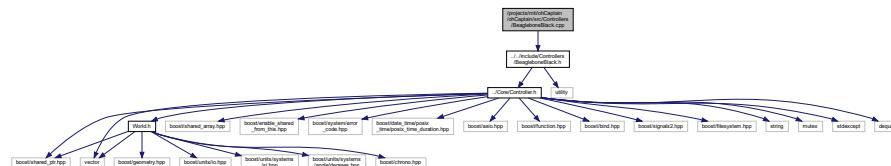


Namespaces

- oCpt
 - oCpt::components
 - oCpt::components::comm

7.21 /projects/mti/ohCaptain/ohCaptain/src/Controllers/BeagleboneBlack.cpp File Reference

```
#include "../../include/Controllers/BeagleboneBlack.h"
Include dependency graph for BeagleboneBlack.cpp:
```

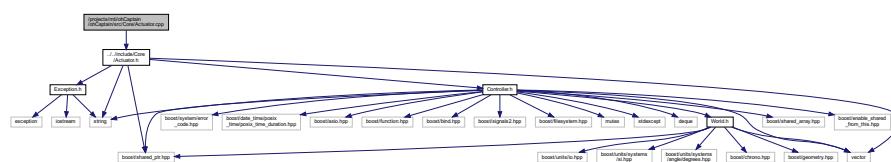


Namespaces

- `oCpt`
- `oCpt::components`
- `oCpt::components::controller`

7.22 /projects/mti/ohCaptain/ohCaptain/src/Core/Actuator.cpp File Reference

```
#include "../../include/Core/Actuator.h"
Include dependency graph for Actuator.cpp:
```

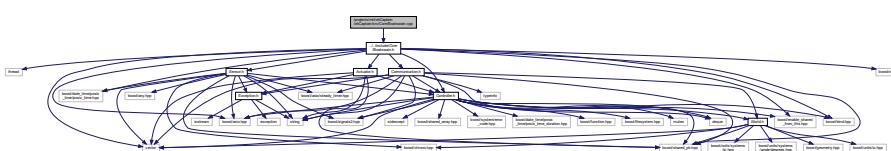


Namespaces

- `oCpt`

7.23 /projects/mti/ohCaptain/ohCaptain/src/Core/Boatswain.cpp File Reference

```
#include "../../include/Core/Boatswain.h"
Include dependency graph for Boatswain.cpp:
```



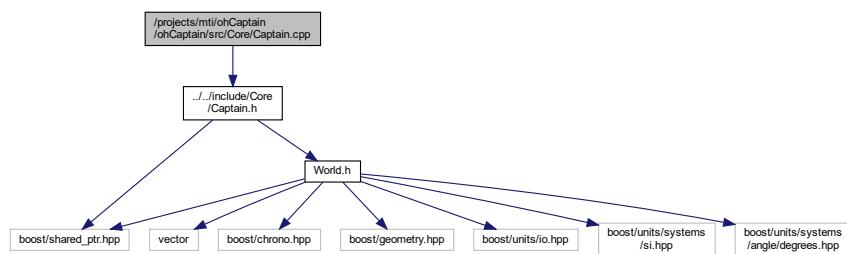
Namespaces

- oCpt

7.24 /projects/mti/ohCaptain/ohCaptain/src/Core/Captain.cpp File Reference

```
#include "../../include/Core/Captain.h"
```

Include dependency graph for Captain.cpp:



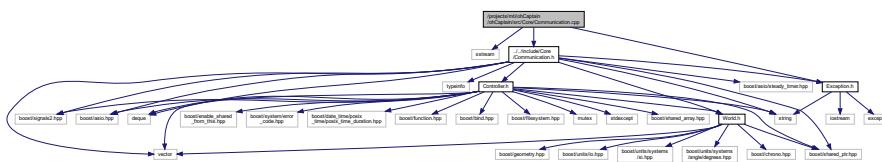
Namespaces

- oCpt

7.25 /projects/mti/ohCaptain/ohCaptain/src/Core/Communication.cpp File Reference

```
#include <iostream>
#include "../../include/Core/Communication.h"
#include "../../include/Core/Exception.h"
```

Include dependency graph for Communication.cpp:

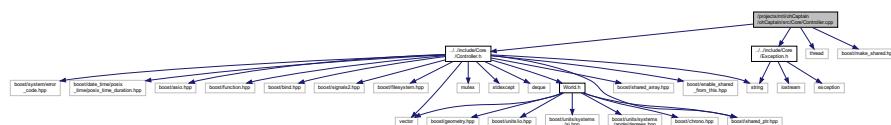


Namespaces

- oCpt

7.26 /projects/mti/ohCaptain/ohCaptain/src/Core/Controller.cpp File Reference

```
#include "../../include/Core/Controller.h"
#include "../../include/Core/Exception.h"
#include <thread>
#include <boost/make_shared.hpp>
Include dependency graph for Controller.cpp:
```

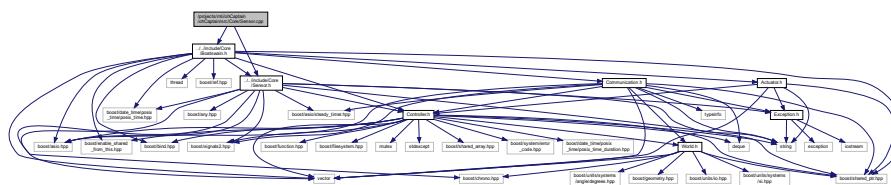


Namespaces

- oCpt
 - oCpt::protocol

7.27 /projects/mti/ohCaptain/ohCaptain/src/Core/Sensor.cpp File Reference

```
#include "../../include/Core/Sensor.h"  
#include "../../include/Core/Boatswain.h"  
Include dependency graph for Sensor.cpp:
```

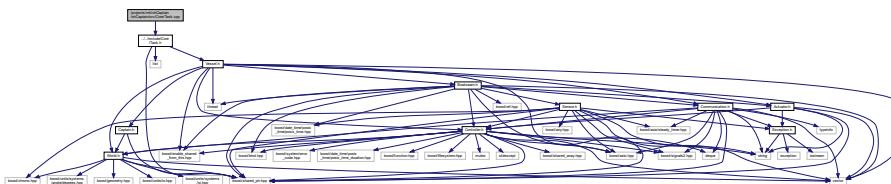


Namespaces

- oCpt

7.28 /projects/mti/ohCaptain/ohCaptain/src/Core/Task.cpp File Reference

```
#include "../..../include/Core/Task.h"
Include dependency graph for Task.cpp:
```

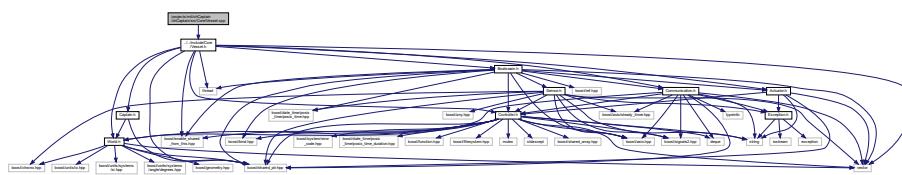


Namespaces

- oCpt

7.29 /projects/mti/ohCaptain/ohCaptain/src/Core/Vessel.cpp File Reference

```
#include "../../include/Core/Vessel.h"
Include dependency graph for Vessel.cpp:
```

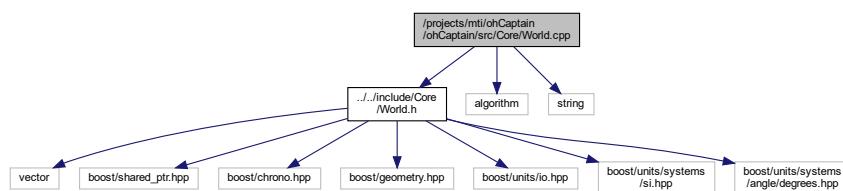


Namespaces

- oCpt

7.30 /projects/mti/ohCaptain/ohCaptain/src/Core/World.cpp File Reference

```
#include "../../include/Core/World.h"
#include <algorithm>
#include <string>
Include dependency graph for World.cpp:
```



Namespaces

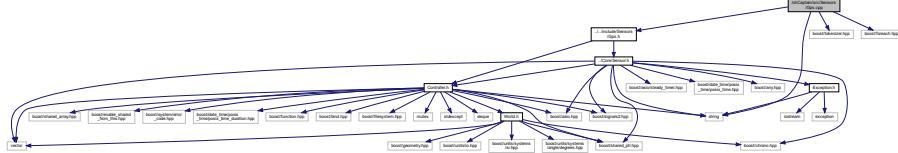
- oCpt

7.31 /projects/mti/ohCaptain/ohCaptain/src/Sensors/Gps.cpp File Reference

```
#include "../../include/Sensors/Gps.h"
#include <boost/tokenizer.hpp>
#include <boost/foreach.hpp>
#include <string>
```

Include dependency graph for Gps.cpp:

• 198 •



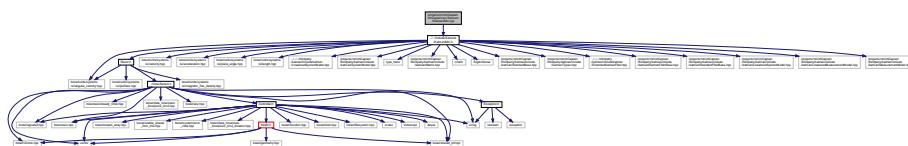
Namespaces

- `oCpt`
 - `oCpt::components`
 - `oCpt::components::sensors`

7.32 /projects/mti/ohCaptain/ohCaptain/src/Sensors/KalmanIMU.cpp File Reference

```
#include "../include/Sensors/KalmanIMU.h"
```

Include dependency graph for KalmanIMU.cpp:



Namespaces

- `oCpt`
 - `oCpt::components`
 - `oCpt::components::sensors`

7.33 /projects/mti/ohCaptain/ohCaptain/src/Sensors/PT100.cpp File Reference

```
#include "../../include/Sensors/PT100.h"
#include "../../include/Core/Sensor.h"
#include <iostream>
```

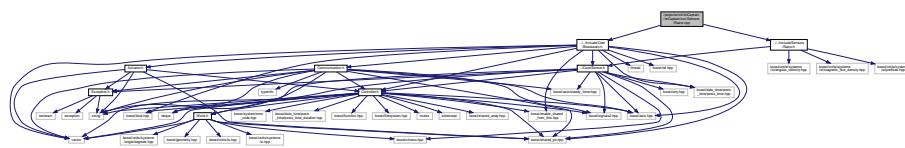
Include dependency graph for PT100.cpp:

Namespaces

- [oCpt](#)
- [oCpt::components](#)
- [oCpt::components::sensors](#)

7.34 /projects/mti/ohCaptain/ohCaptain/src/Sensors/Razor.cpp File Reference

```
#include "../../include/Sensors/Razor.h"
#include "../../include/Core/Boatswain.h"
Include dependency graph for Razor.cpp:
```

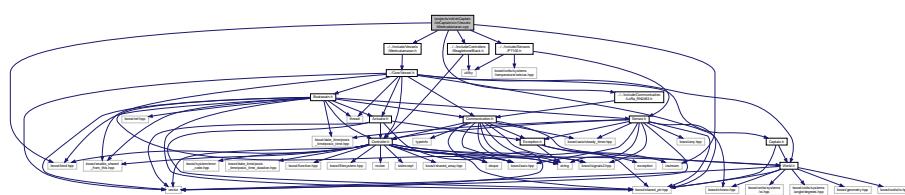


Namespaces

- [oCpt](#)
- [oCpt::components](#)
- [oCpt::components::sensors](#)

7.35 /projects/mti/ohCaptain/ohCaptain/src/Vessels/Meetcatamaran.cpp File Reference

```
#include "../../include/Vessels/Meetcatamaran.h"
#include "../../include/Controllers/BeagleboneBlack.h"
#include "../../include/Sensors/PT100.h"
#include "../../include/Communication/LoRa_RN2483.h"
#include <boost/bind.hpp>
#include <iostream>
Include dependency graph for Meetcatamaran.cpp:
```



Namespaces

- [oCpt](#)
- [oCpt::vessels](#)