

My Qt Project

Generated by Doxygen 1.15.0

1 Directory Hierarchy	1
1.1 Directories	1
2 Namespace Index	5
2.1 Namespace List	5
3 Hierarchical Index	7
3.1 Class Hierarchy	7
4 Data Structure Index	9
4.1 Data Structures	9
5 File Index	11
5.1 File List	11
6 Directory Documentation	15
6.1 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/← Header/driver Directory Reference	15
6.2 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/← Source/driver Directory Reference	16
6.3 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example Directory Reference	16
6.4 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/← Header/generic Directory Reference	17
6.5 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/← Source/generic Directory Reference	18
6.6 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header Directory Reference	19
6.7 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source Directory Reference	20
7 Namespace Documentation	21
7.1 Ui Namespace Reference	21
8 Data Structure Documentation	23
8.1 AltimeterData Struct Reference	23
8.1.1 Field Documentation	23
8.1.1.1 altitude	23
8.1.1.2 pressure	23
8.1.1.3 relative	23
8.1.1.4 temperature	23
8.2 BleUart Class Reference	24
8.2.1 Member Typedef Documentation	25
8.2.1.1 RxCallback	25
8.2.2 Member Function Documentation	25
8.2.2.1 bytesReceived()	25
8.2.2.2 connectTo()	25

8.2.2.3 dataReceived()	25
8.2.2.4 disconnected()	26
8.2.2.5 ready()	26
8.2.2.6 setRxCallback()	26
8.2.2.7 startScan()	26
8.2.2.8 writeBytes()	26
8.2.3 Field Documentation	27
8.2.3.1 scancomplete	27
8.3 BTComQt Class Reference	27
8.3.1 Member Typedef Documentation	28
8.3.1.1 RxCallback	28
8.3.2 Constructor & Destructor Documentation	28
8.3.2.1 BTComQt()	28
8.3.2.2 ~BTComQt()	28
8.3.3 Member Function Documentation	29
8.3.3.1 close()	29
8.3.3.2 dataReceived()	29
8.3.3.3 open()	29
8.3.3.4 send() [1/2]	30
8.3.3.5 send() [2/2]	30
8.3.3.6 setRxCallback()	30
8.3.4 Field Documentation	30
8.3.4.1 imuJavaObject	30
8.3.4.2 serial_	30
8.3.4.3 timerAndroid	31
8.4 MyTcpSocket::Callbacks Struct Reference	31
8.4.1 Field Documentation	31
8.4.1.1 callback	31
8.4.1.2 classPtr	31
8.5 CharacteristicInfo Class Reference	32
8.5.1 Constructor & Destructor Documentation	32
8.5.1.1 CharacteristicInfo() [1/2]	32
8.5.1.2 CharacteristicInfo() [2/2]	33
8.5.2 Member Function Documentation	33
8.5.2.1 characteristicChanged()	33
8.5.2.2 getCharacteristic()	33
8.5.2.3 getName()	33
8.5.2.4 getPermission()	33
8.5.2.5 getUuid()	33
8.5.2.6 getValue()	34
8.5.2.7 setCharacteristic()	34
8.6 ChatClient Class Reference	34

8.6.1 Detailed Description	35
8.6.2 Constructor & Destructor Documentation	35
8.6.2.1 ChatClient()	35
8.6.2.2 ~ChatClient()	36
8.6.3 Member Function Documentation	36
8.6.3.1 connected()	36
8.6.3.2 disconnected()	36
8.6.3.3 messageReceived()	37
8.6.3.4 sendMessage()	37
8.6.3.5 socketErrorOccurred()	37
8.6.3.6 startClient()	37
8.6.3.7 stopClient()	38
8.7 ChatServer Class Reference	38
8.7.1 Detailed Description	39
8.7.2 Constructor & Destructor Documentation	39
8.7.2.1 ChatServer()	39
8.7.2.2 ~ChatServer()	40
8.7.3 Member Function Documentation	40
8.7.3.1 clientConnected()	40
8.7.3.2 clientDisconnected()	40
8.7.3.3 messageReceived()	40
8.7.3.4 sendMessage()	41
8.7.3.5 startServer()	41
8.7.3.6 stopServer()	42
8.8 ComBt Class Reference	42
8.8.1 Member Typedef Documentation	44
8.8.1.1 RxCallback	44
8.8.2 Constructor & Destructor Documentation	44
8.8.2.1 ComBt()	44
8.8.2.2 ~ComBt()	44
8.8.3 Member Function Documentation	45
8.8.3.1 close()	45
8.8.3.2 connectionChanged()	45
8.8.3.3 dataReceived()	45
8.8.3.4 errorReceived()	45
8.8.3.5 open()	46
8.8.3.6 send() [1/2]	46
8.8.3.7 send() [2/2]	46
8.8.3.8 setBaudrate()	46
8.8.3.9 setRxCallback()	47
8.8.4 Field Documentation	47
8.8.4.1 callback_	47

8.8.4.2 parent	47
8.8.4.3 scanDone	47
8.8.4.4 serial_	47
8.8.4.5 status	47
8.9 ComQt Class Reference	48
8.9.1 Detailed Description	49
8.9.2 Member Typedef Documentation	49
8.9.2.1 RxCallback	49
8.9.3 Constructor & Destructor Documentation	50
8.9.3.1 ComQt()	50
8.9.3.2 ~ComQt()	50
8.9.4 Member Function Documentation	50
8.9.4.1 close()	50
8.9.4.2 connectionChanged()	51
8.9.4.3 dataReceived()	51
8.9.4.4 errorReceived()	52
8.9.4.5 isOpen()	52
8.9.4.6 open()	52
8.9.4.7 send() [1/2]	53
8.9.4.8 send() [2/2]	53
8.9.4.9 setBaudrate()	54
8.9.4.10 setRxCallback()	54
8.9.5 Field Documentation	54
8.9.5.1 callback_	54
8.9.5.2 parent	54
8.9.5.3 serial_	54
8.9.5.4 status	55
8.10 ConnListener Class Reference	55
8.10.1 Detailed Description	56
8.10.2 Constructor & Destructor Documentation	56
8.10.2.1 ConnListener()	56
8.10.3 Member Function Documentation	56
8.10.3.1 on_failure()	56
8.10.3.2 on_success()	56
8.11 Device Class Reference	57
8.11.1 Constructor & Destructor Documentation	58
8.11.1.1 Device()	58
8.11.1.2 ~Device()	58
8.11.2 Member Function Documentation	58
8.11.2.1 characteristicsUpdated()	58
8.11.2.2 connectToService()	59
8.11.2.3 disconnected()	59

8.11.2.4 disconnectFromDevice()	59
8.11.2.5 randomAddressChanged()	59
8.11.2.6 scanServices()	59
8.11.2.7 sendData()	60
8.11.2.8 setVal()	60
8.11.2.9 startDeviceDiscovery()	60
8.11.2.10 state()	60
8.11.2.11 stateChanged()	61
8.11.2.12 test()	61
8.11.2.13 updateChanged()	61
8.11.2.14 updateDisplay()	61
8.11.2.15 updateValueSet()	61
8.11.3 Field Documentation	62
8.11.3.1 key2	62
8.11.3.2 l_service	62
8.11.3.3 m_message	62
8.11.3.4 receivedData	62
8.12 DeviceInfo Class Reference	62
8.12.1 Constructor & Destructor Documentation	63
8.12.1.1 DeviceInfo() [1/2]	63
8.12.1.2 DeviceInfo() [2/2]	63
8.12.2 Member Function Documentation	63
8.12.2.1 deviceChanged()	63
8.12.2.2 getAddress()	64
8.12.2.3 getDevice()	64
8.12.2.4 getName()	64
8.12.2.5 setDevice()	64
8.12.3 Property Documentation	64
8.12.3.1 deviceName	64
8.13 GeoidHelper Class Reference	64
8.13.1 Constructor & Destructor Documentation	65
8.13.1.1 GeoidHelper()	65
8.13.2 Member Function Documentation	65
8.13.2.1 compensatedHeight()	65
8.13.2.2 isValid()	65
8.14 GeoidResult Struct Reference	65
8.14.1 Field Documentation	65
8.14.1.1 h_compensated	65
8.14.1.2 N	66
8.15 GpxParser Class Reference	66
8.15.1 Member Function Documentation	66
8.15.1.1 getTrackPoints()	66

8.15.1.2 parseFile()	66
8.16 GeoidHelper::Impl Struct Reference	67
8.16.1 Field Documentation	67
8.16.1.1 valid	67
8.17 KeepAwakeHelper Class Reference	67
8.17.1 Constructor & Destructor Documentation	67
8.17.1.1 KeepAwakeHelper()	67
8.17.1.2 ~KeepAwakeHelper()	67
8.17.2 Member Function Documentation	67
8.17.2.1 EnableKeepAwakeHelper()	67
8.18 LayoutSquare Class Reference	68
8.18.1 Detailed Description	69
8.18.2 Constructor & Destructor Documentation	69
8.18.2.1 LayoutSquare() [1/2]	69
8.18.2.2 LayoutSquare() [2/2]	69
8.18.2.3 ~LayoutSquare()	69
8.18.3 Member Function Documentation	69
8.18.3.1 addItem()	69
8.18.3.2 addWidget()	70
8.18.3.3 count()	70
8.18.3.4 expandingDirections()	70
8.18.3.5 geometry2()	70
8.18.3.6 hasHeightForWidth()	70
8.18.3.7 hasItem()	71
8.18.3.8 itemAt()	71
8.18.3.9 minimumSize()	71
8.18.3.10 replaceItem()	71
8.18.3.11 setGeometry()	72
8.18.3.12 sizeHint()	73
8.18.3.13 take()	73
8.18.3.14 takeAt()	73
8.19 LFD Struct Reference	74
8.19.1 Field Documentation	74
8.19.1.1 D	74
8.19.1.2 F	74
8.19.1.3 L	74
8.20 MainWindow Class Reference	74
8.20.1 Detailed Description	78
8.20.2 Constructor & Destructor Documentation	78
8.20.2.1 MainWindow() [1/2]	78
8.20.2.2 ~MainWindow() [1/2]	79
8.20.2.3 MainWindow() [2/2]	79

8.20.2.4 ~MainWindow() [2/2]	79
8.20.3 Member Function Documentation	80
8.20.3.1 AccelerometerRead()	80
8.20.3.2 addcurrent()	80
8.20.3.3 addnext()	80
8.20.3.4 get_default_config()	80
8.20.3.5 getActiveScreen()	81
8.20.3.6 getAI()	81
8.20.3.7 getALT()	81
8.20.3.8 getASI()	81
8.20.3.9 getBearing()	81
8.20.3.10 getEADI()	82
8.20.3.11 getEHJSI()	82
8.20.3.12 getHI()	82
8.20.3.13 getTC()	82
8.20.3.14 getVSI()	82
8.20.3.15 hideCamera()	82
8.20.3.16 init()	82
8.20.3.17 logLanded()	83
8.20.3.18 logTakeoff()	83
8.20.3.19 onResized()	83
8.20.3.20 permissionUpdated()	83
8.20.3.21 set_default_config()	83
8.20.3.22 set_default_planes()	84
8.20.3.23 set_default_radio()	84
8.20.3.24 setButtonIcon()	85
8.20.3.25 setCamera()	85
8.20.3.26 setIMU()	85
8.20.3.27 setmode()	86
8.20.3.28 setQNH()	86
8.20.3.29 showImage()	86
8.20.3.30 timerEvent()	86
8.20.3.31 updateCameras()	87
8.20.4 Field Documentation	87
8.20.4.1 _IMU_id	87
8.20.4.2 _radar_id	87
8.20.4.3 _transponder_id	87
8.20.4.4 _widgetAI	87
8.20.4.5 _widgetALT	87
8.20.4.6 _widgetASI	87
8.20.4.7 _widgetEADI	88
8.20.4.8 _widgetEHSI	88

8.20.4.9 _widgetHI	88
8.20.4.10 _widgetTC	88
8.20.4.11 _widgetVSI	88
8.20.4.12 current	88
8.20.4.13 currentIndex	88
8.20.4.14 m_attitude	88
8.20.4.15 m_bluetoothrunning	88
8.20.4.16 m_head	89
8.20.4.17 m_install	89
8.20.4.18 m_landedTime	89
8.20.4.19 m_ms	89
8.20.4.20 m_msgBox	89
8.20.4.21 m_roll_angle	89
8.20.4.22 m_takeoffTime	89
8.20.4.23 m_tansALT	89
8.20.4.24 m_temp	90
8.20.4.25 m_timer	90
8.20.4.26 m_total_accel	90
8.20.4.27 m_use_imu	90
8.20.4.28 m_var_speed	90
8.20.4.29 m_vario	90
8.20.4.30 mode	90
8.20.4.31 mysocket	90
8.20.4.32 next	91
8.20.4.33 Radar_depth	91
8.20.4.34 Radar_Height	91
8.20.4.35 roll_blended	91
8.20.4.36 roll_blended_ok	91
8.20.4.37 rotationMatrix	91
8.20.4.38 screen_index	91
8.20.4.39 ScreenMode	91
8.20.4.40 series	92
8.20.4.41 splash	92
8.20.4.42 takeoff_altitude	92
8.20.4.43 takeoff_latitude	92
8.20.4.44 takeoff_longitude	92
8.20.4.45 ui	92
8.20.4.46 view	92
8.21 MqttClient Class Reference	92
8.21.1 Detailed Description	93
8.21.2 Member Typedef Documentation	93
8.21.2.1 MsgHandler	93

8.21.3 Constructor & Destructor Documentation	94
8.21.3.1 MqttClient()	94
8.21.4 Member Function Documentation	94
8.21.4.1 client()	94
8.21.4.2 connect()	94
8.21.4.3 disconnect()	94
8.21.4.4 sendMessage()	95
8.21.4.5 setMessageHandler()	95
8.21.4.6 subscribe()	95
8.22 MulticastListener Class Reference	96
8.22.1 Constructor & Destructor Documentation	96
8.22.1.1 MulticastListener()	96
8.22.1.2 ~MulticastListener()	97
8.22.2 Member Function Documentation	97
8.22.2.1 errorOccurred()	97
8.22.2.2 messageReceived()	97
8.22.2.3 start()	98
8.22.2.4 stop()	98
8.23 MyDevice Class Reference	98
8.23.1 Constructor & Destructor Documentation	99
8.23.1.1 MyDevice()	99
8.24 MyTcpSocket Class Reference	99
8.24.1 Detailed Description	103
8.24.2 Member Typedef Documentation	103
8.24.2.1 Callbacks	103
8.24.3 Constructor & Destructor Documentation	104
8.24.3.1 MyTcpSocket()	104
8.24.3.2 ~MyTcpSocket()	105
8.24.4 Member Function Documentation	105
8.24.4.1 CallSwiftMemberFromC()	105
8.24.4.2 com_setup()	105
8.24.4.3 connected()	106
8.24.4.4 connectedAltitude()	106
8.24.4.5 connectedIMU()	107
8.24.4.6 connectedRadar()	108
8.24.4.7 doConnect()	109
8.24.4.8 doRadar()	109
8.24.4.9 doStart()	110
8.24.4.10 doTransponder()	111
8.24.4.11 handleUpdate()	112
8.24.4.12 parseAltimeterLine()	112
8.24.4.13 parseIMU()	113

8.24.4.14 readyWrite()	113
8.24.4.15 sendMessage()	114
8.24.4.16 setbacklit()	114
8.24.4.17 setSerialPorts()	115
8.24.4.18 ssdpConfig()	115
8.24.5 Field Documentation	116
8.24.5.1 _IMU_id	116
8.24.5.2 _radar_id	116
8.24.5.3 _transponder_id	116
8.24.5.4 AccX	116
8.24.5.5 AccY	116
8.24.5.6 AccZ	116
8.24.5.7 Altimeter_data	116
8.24.5.8 Altitudestat	116
8.24.5.9 AngleX	117
8.24.5.10 AngleY	117
8.24.5.11 AngleZ	117
8.24.5.12 AsX	117
8.24.5.13 AsY	117
8.24.5.14 AsZ	117
8.24.5.15 callbacks	117
8.24.5.16 CLIENT_ID	117
8.24.5.17 disc	118
8.24.5.18 Donwn_Speed	118
8.24.5.19 FromID	118
8.24.5.20 G	118
8.24.5.21 HX	118
8.24.5.22 HY	118
8.24.5.23 HZ	118
8.24.5.24 IMUconnected	118
8.24.5.25 imuData	119
8.24.5.26 lidar	119
8.24.5.27 m_address	119
8.24.5.28 m_altitude	119
8.24.5.29 m_external	119
8.24.5.30 m_gpsbearing	119
8.24.5.31 m_gpsspeed	119
8.24.5.32 m_has_MQTT	119
8.24.5.33 m_has_MQTT_accel	120
8.24.5.34 m_has_MQTT_airspeed	120
8.24.5.35 m_has_MQTT_gyro	120
8.24.5.36 m_has_MQTT_heading	120

8.24.5.37 m_has_MQTT_presssure	120
8.24.5.38 m_has_MQTT_vsi	120
8.24.5.39 m_imu_setup_done	120
8.24.5.40 m_latitude	120
8.24.5.41 m_longitude	120
8.24.5.42 m_pressre	121
8.24.5.43 m_pressre_alt	121
8.24.5.44 m_pressre_QNH	121
8.24.5.45 m_pressure_raw	121
8.24.5.46 m_speed	121
8.24.5.47 m_vel_active	121
8.24.5.48 m_vel_D	121
8.24.5.49 m_vel_E	121
8.24.5.50 m_vel_N	122
8.24.5.51 MCAST_PORT	122
8.24.5.52 mqtt	122
8.24.5.53 Orient	122
8.24.5.54 port	122
8.24.5.55 Radarstat	122
8.24.5.56 rDist	122
8.24.5.57 ret_imu	122
8.24.5.58 ret_transponder	123
8.24.5.59 rPos	123
8.24.5.60 rSpeed	123
8.24.5.61 serialport	123
8.24.5.62 SERVER_ADDRESS	123
8.24.5.63 sport	123
8.24.5.64 Temp	123
8.24.5.65 text	123
8.24.5.66 Transponderstat	124
8.24.5.67 TransponderstatWithBarometer	124
8.24.5.68 use_ins_only	124
8.24.5.69 VER	124
8.25 NoButtonMessageBox Class Reference	124
8.25.1 Detailed Description	125
8.25.2 Constructor & Destructor Documentation	125
8.25.2.1 NoButtonMessageBox()	125
8.25.3 Member Function Documentation	125
8.25.3.1 setText()	125
8.26 PortEntry Struct Reference	126
8.26.1 Detailed Description	126
8.26.2 Field Documentation	126

8.26.2.1 description	126
8.26.2.2 manufacturer	127
8.26.2.3 portName	127
8.26.2.4 productId	127
8.26.2.5 serial	127
8.26.2.6 systemLocation	127
8.26.2.7 vendorId	127
8.27 QuantumCircuit Class Reference	127
8.27.1 Constructor & Destructor Documentation	128
8.27.1.1 QuantumCircuit() [1/2]	128
8.27.1.2 QuantumCircuit() [2/2]	129
8.27.2 Member Function Documentation	129
8.27.2.1 add()	129
8.27.2.2 ch()	130
8.27.2.3 crx()	130
8.27.2.4 cx()	130
8.27.2.5 h()	130
8.27.2.6 has_measurements()	131
8.27.2.7 initialize()	131
8.27.2.8 measure()	131
8.27.2.9 rx()	131
8.27.2.10 ry()	132
8.27.2.11 rz()	132
8.27.2.12 set_registers()	133
8.27.2.13 x()	133
8.27.2.14 y()	134
8.27.2.15 z()	134
8.27.3 Field Documentation	134
8.27.3.1 data	134
8.27.3.2 nBits	135
8.27.3.3 nQubits	135
8.28 RemoteSelector Class Reference	135
8.28.1 Constructor & Destructor Documentation	136
8.28.1.1 RemoteSelector()	136
8.28.1.2 ~RemoteSelector()	136
8.28.2 Member Function Documentation	137
8.28.2.1 service()	137
8.28.2.2 startDiscovery()	137
8.28.2.3 stopDiscovery()	137
8.29 SensorService Struct Reference	137
8.29.1 Field Documentation	137
8.29.1.1 name	137

8.29.1.2 <code>st</code>	137
8.29.1.3 <code>uuid</code>	138
8.30 ServiceInfo Class Reference	138
8.30.1 Constructor & Destructor Documentation	139
8.30.1.1 <code>ServiceInfo()</code> [1/2]	139
8.30.1.2 <code>ServiceInfo()</code> [2/2]	139
8.30.2 Member Function Documentation	139
8.30.2.1 <code>getName()</code>	139
8.30.2.2 <code>getType()</code>	139
8.30.2.3 <code>getUuid()</code>	139
8.30.2.4 <code>service()</code>	140
8.30.2.5 <code>serviceChanged()</code>	140
8.31 Simulator Class Reference	140
8.31.1 Constructor & Destructor Documentation	141
8.31.1.1 <code>Simulator()</code>	141
8.31.2 Member Function Documentation	141
8.31.2.1 <code>get_counts()</code>	141
8.31.2.2 <code>get_memory()</code>	142
8.31.2.3 <code>get_qasm()</code>	142
8.31.2.4 <code>get_qiskit()</code>	142
8.31.2.5 <code>get_statevector()</code>	143
8.31.3 Field Documentation	143
8.31.3.1 <code>qc</code>	143
8.31.3.2 <code>shots</code>	143
8.32 SsdpDiscoverer Class Reference	143
8.32.1 Constructor & Destructor Documentation	144
8.32.1.1 <code>SsdpDiscoverer()</code>	144
8.32.2 Member Function Documentation	145
8.32.2.1 <code>deviceFound()</code>	145
8.32.2.2 <code>startDiscovery()</code>	145
8.33 TcpClient Class Reference	146
8.33.1 Constructor & Destructor Documentation	147
8.33.1.1 <code>TcpClient()</code>	147
8.33.2 Member Function Documentation	147
8.33.2.1 <code>connected()</code>	147
8.33.2.2 <code>connectTo()</code>	147
8.33.2.3 <code>dataReceived()</code>	147
8.33.2.4 <code>disconnected()</code>	147
8.33.2.5 <code>disconnectFrom()</code>	147
8.33.2.6 <code>errorOccurred()</code>	148
8.33.2.7 <code>sendData()</code>	148
8.34 TrackPoint Struct Reference	148

8.34.1 Field Documentation	148
8.34.1.1 dt	148
8.34.1.2 elevation	148
8.34.1.3 latitude	148
8.34.1.4 longitude	149
8.34.1.5 speed	149
8.34.1.6 time	149
8.35 WidgetAI Class Reference	149
8.35.1 Constructor & Destructor Documentation	150
8.35.1.1 WidgetAI()	150
8.35.1.2 ~WidgetAI()	150
8.35.2 Member Function Documentation	151
8.35.2.1 redraw()	151
8.35.2.2 reinit()	151
8.35.2.3 setPitch()	151
8.35.2.4 setRoll()	151
8.36 WidgetALT Class Reference	151
8.36.1 Constructor & Destructor Documentation	152
8.36.1.1 WidgetALT()	152
8.36.1.2 ~WidgetALT()	153
8.36.2 Member Function Documentation	153
8.36.2.1 redraw()	153
8.36.2.2 reinit()	153
8.36.2.3 setAltitude()	153
8.36.2.4 setPressure()	153
8.37 WidgetASI Class Reference	154
8.37.1 Constructor & Destructor Documentation	154
8.37.1.1 WidgetASI()	154
8.37.1.2 ~WidgetASI()	155
8.37.2 Member Function Documentation	155
8.37.2.1 redraw()	155
8.37.2.2 reinit()	155
8.37.2.3 setAirspeed()	155
8.38 WidgetEADI Class Reference	156
8.38.1 Constructor & Destructor Documentation	157
8.38.1.1 WidgetEADI()	157
8.38.1.2 ~WidgetEADI()	157
8.38.2 Member Function Documentation	158
8.38.2.1 redraw()	158
8.38.2.2 reinit()	158
8.38.2.3 setAirspeed()	158
8.38.2.4 setAirspeedSel()	158

8.38.2.5 setAltitude()	158
8.38.2.6 setAltitudeSel()	158
8.38.2.7 setClimbRate()	158
8.38.2.8 setDots()	159
8.38.2.9 setFD()	159
8.38.2.10 setFltMode()	159
8.38.2.11 setFPM()	159
8.38.2.12 setHeading()	159
8.38.2.13 setHeadingSel()	160
8.38.2.14 setLNAV()	160
8.38.2.15 setMachNo()	160
8.38.2.16 setPitch()	160
8.38.2.17 setPressure()	160
8.38.2.18 setRoll()	160
8.38.2.19 setSlipSkid()	160
8.38.2.20 setSpdMode()	160
8.38.2.21 setStall()	161
8.38.2.22 setTurnRate()	161
8.38.2.23 setVfe()	161
8.38.2.24 setVNAV()	161
8.38.2.25 setVne()	161
8.39 WidgetEHSI Class Reference	162
8.39.1 Constructor & Destructor Documentation	163
8.39.1.1 WidgetEHSI()	163
8.39.1.2 ~WidgetEHSI()	163
8.39.2 Member Function Documentation	163
8.39.2.1 redraw()	163
8.39.2.2 reinit()	163
8.39.2.3 setBearing()	164
8.39.2.4 setCourse()	164
8.39.2.5 setDeviation()	164
8.39.2.6 setDistance()	164
8.39.2.7 setHeading()	164
8.39.2.8 setHeadingSel()	164
8.40 WidgetHI Class Reference	165
8.40.1 Constructor & Destructor Documentation	165
8.40.1.1 WidgetHI()	165
8.40.1.2 ~WidgetHI()	166
8.40.2 Member Function Documentation	166
8.40.2.1 redraw()	166
8.40.2.2 reinit()	166
8.40.2.3 setHeading()	166

8.41 WidgetSix Class Reference	167
8.41.1 Constructor & Destructor Documentation	168
8.41.1.1 WidgetSix()	168
8.41.1.2 ~WidgetSix()	168
8.41.2 Member Function Documentation	168
8.41.2.1 getAI()	168
8.41.2.2 getALT()	168
8.41.2.3 getASI()	169
8.41.2.4 getEADI()	169
8.41.2.5 getHI()	169
8.41.2.6 getTC()	169
8.41.2.7 getVSI()	169
8.42 WidgetTC Class Reference	169
8.42.1 Constructor & Destructor Documentation	170
8.42.1.1 WidgetTC()	170
8.42.1.2 ~WidgetTC()	171
8.42.2 Member Function Documentation	171
8.42.2.1 redraw()	171
8.42.2.2 reinit()	171
8.42.2.3 setSlipSkid()	171
8.42.2.4 setTurnRate()	171
8.43 WidgetVSI Class Reference	172
8.43.1 Constructor & Destructor Documentation	172
8.43.1.1 WidgetVSI()	172
8.43.1.2 ~WidgetVSI()	173
8.43.2 Member Function Documentation	173
8.43.2.1 redraw()	173
8.43.2.2 reinit()	173
8.43.2.3 setClimbRate()	173
9 File Documentation	175
9.1 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/← LayoutSquare.cpp File Reference	175
9.1.1 Function Documentation	175
9.1.1.1 margin()	175
9.2 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/← LayoutSquare.h File Reference	176
9.3 LayoutSquare.h	176
9.4 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/← WidgetAI.cpp File Reference	178
9.5 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/← WidgetAI.h File Reference	178
9.6 WidgetAI.h	179

9.7 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/WidgetALT.cpp File Reference	180
9.8 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/WidgetALT.h File Reference	181
9.9 WidgetALT.h	182
9.10 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/WidgetASI.cpp File Reference	183
9.11 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/WidgetASI.h File Reference	183
9.12 WidgetASI.h	184
9.13 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/WidgetEADI.cpp File Reference	185
9.14 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/WidgetEADI.h File Reference	186
9.15 WidgetEADI.h	187
9.16 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/WidgetEHSI.cpp File Reference	189
9.17 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/WidgetEHSI.h File Reference	189
9.18 WidgetEHSI.h	191
9.19 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/WidgetHI.cpp File Reference	192
9.20 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/WidgetHI.h File Reference	192
9.21 WidgetHI.h	194
9.22 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/WidgetSix.cpp File Reference	194
9.23 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/WidgetSix.h File Reference	195
9.24 WidgetSix.h	196
9.25 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/WidgetTC.cpp File Reference	197
9.26 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/WidgetTC.h File Reference	198
9.27 WidgetTC.h	199
9.28 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/WidgetVSI.cpp File Reference	200
9.29 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/WidgetVSI.h File Reference	200
9.30 WidgetVSI.h	201
9.31 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header/driver/bleuart.h File Reference	202
9.31.1 Function Documentation	203
9.31.1.1 wt901Rx()	203
9.31.1.2 wt901Service()	203
9.31.1.3 wt901Tx()	204
9.32 bleuart.h	204

9.33	/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/↔ Header/driver/geoid_helper.h File Reference	207
9.34	geoid_helper.h	208
9.35	/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/↔ Header/driver/gpx_parse.h File Reference	208
9.36	gpx_parse.h	209
9.37	/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/↔ Header/driver/mqttclient.h File Reference	209
9.38	mqttclient.h	210
9.39	/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/↔ Header/driver/rotation_matrix.h File Reference	212
9.39.1	Macro Definition Documentation	213
9.39.1.1	DEG_TO_RAD	213
9.39.1.2	M_PI	214
9.39.1.3	RAD_TO_DEG	214
9.39.2	Typedef Documentation	214
9.39.2.1	Matrix3x3	214
9.39.2.2	Matrix3x6	214
9.39.2.3	Matrix9x9	214
9.39.2.4	Vector3x	214
9.39.2.5	Vector9	214
9.39.3	Function Documentation	215
9.39.3.1	createRotationMatrix()	215
9.39.3.2	createRotationMatrixZ()	215
9.39.3.3	multiplyMatrix()	215
9.39.3.4	printVector()	216
9.39.3.5	rotateSensors() [1/3]	216
9.39.3.6	rotateSensors() [2/3]	216
9.39.3.7	rotateSensors() [3/3]	216
9.39.3.8	rotateVector()	217
9.40	rotation_matrix.h	217
9.41	/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/↔ Header/driver/serialport.h File Reference	218
9.42	serialport.h	218
9.43	/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/↔ Header/driver/serialport_bluetooth.h File Reference	219
9.44	serialport_bluetooth.h	220
9.45	/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/↔ Header/driver/wit_c_sdk.h File Reference	221
9.45.1	Detailed Description	223
9.45.2	Macro Definition Documentation	223
9.45.2.1	SCANF_INT	223
9.45.2.2	SLEEP_MS	224
9.45.2.3	UINT32	224

9.45.2.4 WIT_DATA_BUFF_SIZE	224
9.45.2.5 WIT_HAL_BUSY	224
9.45.2.6 WIT_HAL_EMPTY	224
9.45.2.7 WIT_HAL_ERROR	224
9.45.2.8 WIT_HAL_INVAL	224
9.45.2.9 WIT_HAL_NOMEM	224
9.45.2.10 WIT_HAL_OK	225
9.45.2.11 WIT_HAL_TIMEOUT	225
9.45.2.12 WIT_PROTOCOL_905x_CAN	225
9.45.2.13 WIT_PROTOCOL_905x_MODBUS	225
9.45.2.14 WIT_PROTOCOL_CAN	225
9.45.2.15 WIT_PROTOCOL_I2C	225
9.45.2.16 WIT_PROTOCOL_JY61	225
9.45.2.17 WIT_PROTOCOL_MODBUS	225
9.45.2.18 WIT_PROTOCOL_NORMAL	225
9.45.3 Typedef Documentation	226
9.45.3.1 DelaymsCb	226
9.45.3.2 RegUpdateCb	226
9.45.3.3 SerialWrite	226
9.45.4 Function Documentation	226
9.45.4.1 CheckRange()	226
9.45.4.2 posix_delay_ms()	227
9.45.4.3 WitCaliRefAngle()	227
9.45.4.4 WitDeInit()	227
9.45.4.5 WitDelayMsRegister()	227
9.45.4.6 WitInit()	228
9.45.4.7 WitReadReg()	228
9.45.4.8 WitRegisterCallBack()	228
9.45.4.9 WitSaveParameter()	229
9.45.4.10 WitSerialDataIn()	229
9.45.4.11 WitSerialWriteRegister()	230
9.45.4.12 WitSetBandwidth()	230
9.45.4.13 WitSetCanBaud()	230
9.45.4.14 WitSetContent()	231
9.45.4.15 WitSetForReset()	231
9.45.4.16 WitSetOutputRate()	231
9.45.4.17 WitSetUartBaud()	232
9.45.4.18 WitStartAccCali()	232
9.45.4.19 WitStartMagCali()	233
9.45.4.20 WitStopAccCali()	233
9.45.4.21 WitStopMagCali()	233
9.45.4.22 WitWriteReg()	234

9.45.5 Variable Documentation	234
9.45.5.1 sReg	234
9.46 wit_c_sdk.h	235
9.47 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/← Header/generic/characteristicinfo.h File Reference	236
9.48 characteristicinfo.h	236
9.49 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/← Header/generic/chatclient.h File Reference	237
9.50 chatclient.h	238
9.51 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/← Header/generic/chatserver.h File Reference	239
9.52 chatserver.h	240
9.53 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/← Header/generic/device.h File Reference	241
9.54 device.h	242
9.55 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/← Header/generic/deviceinfo.h File Reference	243
9.56 deviceinfo.h	244
9.57 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/← Header/generic/lockhelper.h File Reference	245
9.58 lockhelper.h	246
9.59 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/← MainWindow.h File Reference	247
9.60 MainWindow.h	248
9.61 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/← Header/generic/mainwindow.h File Reference	249
9.61.1 Macro Definition Documentation	250
9.61.1.1 AIRPLANE	250
9.61.1.2 CONFIG	250
9.61.1.3 FLIGHTLOG	250
9.61.1.4 IMAGES_DIR	250
9.61.1.5 LOG_DIR	251
9.61.1.6 P_AUTOPILOT	251
9.61.1.7 P_CAMERA	251
9.61.1.8 P_CONFIG	251
9.61.1.9 P_FLIGT_INSTRUMENT	251
9.61.1.10 P_GLASS_COCPIT	251
9.61.1.11 P_IMU	251
9.61.1.12 P_RADAR	251
9.61.1.13 P_RADIO_LIST	251
9.61.1.14 P_TRANSPOUNDER	251
9.61.1.15 RADIO	252
9.61.1.16 TRANSPONDERLOG	252
9.61.1.17 USE_KeepAwakeHelper	252

9.61.2 Function Documentation	252
9.61.2.1 Qiskit()	252
9.62 mainwindow.h	253
9.63 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/← Header/generic/MicroQiskitCpp.h File Reference	258
9.63.1 Macro Definition Documentation	260
9.63.1.1 ERROR	260
9.63.1.2 Qt	260
9.63.1.3 RED	260
9.63.1.4 RESET	260
9.63.2 Function Documentation	260
9.63.2.1 error_handler()	260
9.64 MicroQiskitCpp.h	260
9.65 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/← Header/generic/multicastlistner.h File Reference	266
9.66 multicastlistner.h	267
9.67 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/← Header/generic/MyDevice.h File Reference	268
9.68 MyDevice.h	268
9.69 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/← Header/generic/myNativeWrapperFunctions.h File Reference	269
9.69.1 Function Documentation	269
9.69.1.1 setlosParams()	269
9.70 myNativeWrapperFunctions.h	270
9.71 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/← Header/generic/mytcpsocket.h File Reference	270
9.71.1 Macro Definition Documentation	271
9.71.1.1 SCREEN	271
9.71.1.2 simGPS	271
9.71.1.3 USE_BT_IMU	272
9.71.2 Variable Documentation	272
9.71.2.1 ETX	272
9.71.2.2 STX	272
9.72 mytcpsocket.h	272
9.73 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/← Header/generic/REG.h File Reference	277
9.73.1 Macro Definition Documentation	281
9.73.1.1 ACCFILT	281
9.73.1.2 ACCRANGE	282
9.73.1.3 ACCSENSOR	282
9.73.1.4 ALARMLEVEL	282
9.73.1.5 ALARMPIN	282
9.73.1.6 ALGRITHM6	282
9.73.1.7 ALGRITHM9	282

9.73.1.8 ANGLEAXIS	282
9.73.1.9 ANGXOFFSET	282
9.73.1.10 ANGYOFFSET	282
9.73.1.11 ANGZOFFSET	282
9.73.1.12 AX	283
9.73.1.13 AXIS6	283
9.73.1.14 AXOFFSET	283
9.73.1.15 AY	283
9.73.1.16 AYOFFSET	283
9.73.1.17 AZ	283
9.73.1.18 AZOFFSET	283
9.73.1.19 BANDWIDTH	283
9.73.1.20 BANDWIDTH_10HZ	283
9.73.1.21 BANDWIDTH_184HZ	283
9.73.1.22 BANDWIDTH_21HZ	284
9.73.1.23 BANDWIDTH_256HZ	284
9.73.1.24 BANDWIDTH_44HZ	284
9.73.1.25 BANDWIDTH_5HZ	284
9.73.1.26 BANDWIDTH_94HZ	284
9.73.1.27 BATVAL	284
9.73.1.28 BAUD	284
9.73.1.29 BWSCALE	284
9.73.1.30 CALACCL	284
9.73.1.31 CALACCR	284
9.73.1.32 CALALTITUDE	285
9.73.1.33 CALANGLEZ	285
9.73.1.34 CALGYROACC	285
9.73.1.35 CALHEXAHEDRON	285
9.73.1.36 CALICNT	285
9.73.1.37 CALMAG	285
9.73.1.38 CALMAG2STEP	285
9.73.1.39 CALMAGMM	285
9.73.1.40 CALREFANGLE	285
9.73.1.41 CALSW	285
9.73.1.42 CAN_BAUD_10000	286
9.73.1.43 CAN_BAUD_100000	286
9.73.1.44 CAN_BAUD_1000000	286
9.73.1.45 CAN_BAUD_125000	286
9.73.1.46 CAN_BAUD_20000	286
9.73.1.47 CAN_BAUD_200000	286
9.73.1.48 CAN_BAUD_250000	286
9.73.1.49 CAN_BAUD_3000	286

9.73.1.50 CAN_BAUD_40000	286
9.73.1.51 CAN_BAUD_400000	286
9.73.1.52 CAN_BAUD_5000	287
9.73.1.53 CAN_BAUD_50000	287
9.73.1.54 CAN_BAUD_500000	287
9.73.1.55 CAN_BAUD_80000	287
9.73.1.56 CAN_BAUD_800000	287
9.73.1.57 CCBW	287
9.73.1.58 CHIPIDH	287
9.73.1.59 CHIPIDL	287
9.73.1.60 D0MODE	287
9.73.1.61 D0PWMH	287
9.73.1.62 D0PWMT	288
9.73.1.63 D0Status	288
9.73.1.64 D1MODE	288
9.73.1.65 D1PWMH	288
9.73.1.66 D1PWMT	288
9.73.1.67 D1Status	288
9.73.1.68 D2MODE	288
9.73.1.69 D2PWMH	288
9.73.1.70 D2PWMT	288
9.73.1.71 D2Status	288
9.73.1.72 D3MODE	289
9.73.1.73 D3PWMH	289
9.73.1.74 D3PWMT	289
9.73.1.75 D3Status	289
9.73.1.76 DDHH	289
9.73.1.77 DELAYT	289
9.73.1.78 FILTK	289
9.73.1.79 GPSBAUD	289
9.73.1.80 GPSHeight	289
9.73.1.81 GPSTYPE	289
9.73.1.82 GPSVH	290
9.73.1.83 GPSVL	290
9.73.1.84 GPSVX_H	290
9.73.1.85 GPSVX_L	290
9.73.1.86 GPSVY_H	290
9.73.1.87 GPSVY_L	290
9.73.1.88 GPSVZ_H	290
9.73.1.89 GPSVZ_L	290
9.73.1.90 GPSYAW	290
9.73.1.91 GX	290

9.73.1.92 GOFFSET	291
9.73.1.93 GY	291
9.73.1.94 GYOFFSET	291
9.73.1.95 GYROCALITHR	291
9.73.1.96 GYROCALTIME	291
9.73.1.97 GYROFILT	291
9.73.1.98 GYRORANGE	291
9.73.1.99 GYROSENSOR	291
9.73.1.100 GYROZSCALE	291
9.73.1.101 GZ	291
9.73.1.102 GZOFFSET	292
9.73.1.103 HDOP	292
9.73.1.104 HeightH	292
9.73.1.105 HeightL	292
9.73.1.106 HPitch	292
9.73.1.107 HRoll	292
9.73.1.108 HXi	292
9.73.1.109 HXOFFSET	292
9.73.1.110 HYaw	292
9.73.1.111 HYi	292
9.73.1.112 HYOFFSET	293
9.73.1.113 HZi	293
9.73.1.114 HZOFFSET	293
9.73.1.115 IICADDR	293
9.73.1.116 KEY	293
9.73.1.117 KEY_UNLOCK	293
9.73.1.118 LatH	293
9.73.1.119 LatL	293
9.73.1.120 LEDOFF	293
9.73.1.121 LonH	293
9.73.1.122 LonL	294
9.73.1.123 LPitch	294
9.73.1.124 LRoll	294
9.73.1.125 LYaw	294
9.73.1.126 MAGFILT	294
9.73.1.127 MAGRANGX	294
9.73.1.128 MAGRANGY	294
9.73.1.129 MAGRANGZ	294
9.73.1.130 MAGSENSOR	294
9.73.1.131 MMSS	294
9.73.1.132 MODDELAY	295
9.73.1.133 MOVESTA	295

9.73.1.134 MOVETHR	295
9.73.1.135 MS	295
9.73.1.136 NORMAL	295
9.73.1.137 NUMBERID1	295
9.73.1.138 NUMBERID2	295
9.73.1.139 NUMBERID3	295
9.73.1.140 NUMBERID4	295
9.73.1.141 NUMBERID5	295
9.73.1.142 NUMBERID6	296
9.73.1.143 ORIENT	296
9.73.1.144 ORIENT_HERIZONE	296
9.73.1.145 ORIENT_VERTICLE	296
9.73.1.146 PDOP	296
9.73.1.147 Pitch	296
9.73.1.148 POWONSEND	296
9.73.1.149 PRESSENSOR	296
9.73.1.150 PressureH	296
9.73.1.151 PressureL	296
9.73.1.152 q0	297
9.73.1.153 q1	297
9.73.1.154 q2	297
9.73.1.155 q3	297
9.73.1.156 READADDR	297
9.73.1.157 REFPITCH	297
9.73.1.158 REFROLL	297
9.73.1.159 REFYAW	297
9.73.1.160 REGINITFLAG	297
9.73.1.161 REGSIZE	297
9.73.1.162 Roll	298
9.73.1.163 RRATE	298
9.73.1.164 RRATE_02HZ	298
9.73.1.165 RRATE_05HZ	298
9.73.1.166 RRATE_100HZ	298
9.73.1.167 RRATE_10HZ	298
9.73.1.168 RRATE_125HZ	298
9.73.1.169 RRATE_1HZ	298
9.73.1.170 RRATE_200HZ	298
9.73.1.171 RRATE_20HZ	298
9.73.1.172 RRATE_2HZ	299
9.73.1.173 RRATE_50HZ	299
9.73.1.174 RRATE_5HZ	299
9.73.1.175 RRATE_NONE	299

9.73.1.176 RRATE_ONCE	299
9.73.1.177 RSW	299
9.73.1.178 RSW_ACC	299
9.73.1.179 RSW_ANGLE	299
9.73.1.180 RSW_GPS	299
9.73.1.181 RSW_GSA	299
9.73.1.182 RSW_GYRO	300
9.73.1.183 RSW_MAG	300
9.73.1.184 RSW_MASK	300
9.73.1.185 RSW_PORT	300
9.73.1.186 RSW_PRESS	300
9.73.1.187 RSW_Q	300
9.73.1.188 RSW_TIME	300
9.73.1.189 RSW_V	300
9.73.1.190 SAVE	300
9.73.1.191 SAVE_PARAM	300
9.73.1.192 SAVE_SWRST	301
9.73.1.193 SLEEP	301
9.73.1.194 SVNUM	301
9.73.1.195 TEMP	301
9.73.1.196 TEMP905x	301
9.73.1.197 TIMEZONE	301
9.73.1.198 TRIGTIME	301
9.73.1.199 VDOP	301
9.73.1.200 VERSION	301
9.73.1.201 WERROR	301
9.73.1.202 WIT_ACC	302
9.73.1.203 WIT_ANGLE	302
9.73.1.204 WIT_BAUD_115200	302
9.73.1.205 WIT_BAUD_19200	302
9.73.1.206 WIT_BAUD_230400	302
9.73.1.207 WIT_BAUD_38400	302
9.73.1.208 WIT_BAUD_460800	302
9.73.1.209 WIT_BAUD_4800	302
9.73.1.210 WIT_BAUD_57600	302
9.73.1.211 WIT_BAUD_921600	302
9.73.1.212 WIT_BAUD_9600	303
9.73.1.213 WIT_DPORT	303
9.73.1.214 WIT_GPS	303
9.73.1.215 WIT_GSA	303
9.73.1.216 WIT_GYRO	303
9.73.1.217 WIT_MAGNETIC	303

9.73.1.218 WIT_PRESS	303
9.73.1.219 WIT_QUATER	303
9.73.1.220 WIT_REGVALUE	303
9.73.1.221 WIT_TIME	303
9.73.1.222 WIT_VELOCITY	304
9.73.1.223 WZCNT	304
9.73.1.224 WZSTATIC	304
9.73.1.225 WZTIME	304
9.73.1.226 XA30NSCALE	304
9.73.1.227 XA30PSCALE	304
9.73.1.228 XA85NSCALE	304
9.73.1.229 XA85PSCALE	304
9.73.1.230 XMAX	304
9.73.1.231 XMIN	304
9.73.1.232 XREFROLL	305
9.73.1.233 XRSCALE	305
9.73.1.234 YA30NSCALE	305
9.73.1.235 YA30PSCALE	305
9.73.1.236 YA85NSCALE	305
9.73.1.237 YA85PSCALE	305
9.73.1.238 Yaw	305
9.73.1.239 YMAX	305
9.73.1.240 YMIN	305
9.73.1.241 YREFPITCH	305
9.73.1.242 YRSCALE	306
9.73.1.243 YYMM	306
9.73.1.244 ZREFYAW	306
9.73.1.245 ZRSCALE	306
9.74 REG.h	306
9.75 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/← Header/generic/remoteselect.h File Reference	310
9.76 remoteselect.h	311
9.77 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/← Header/generic/serviceinfo.h File Reference	312
9.78 serviceinfo.h	312
9.79 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/← Header/generic/ssdp.h File Reference	313
9.79.1 Detailed Description	315
9.79.2 Macro Definition Documentation	315
9.79.2.1 SENSOR	315
9.79.2.2 SENSOR_TYPE	315
9.79.3 Function Documentation	315
9.79.3.1 SSDP_GROUP()	315

9.79.4 Variable Documentation	315
9.79.4.1 SSDP_PORT	315
9.79.4.2 SSDP_ST	316
9.80 ssdp.h	316
9.81 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/← Header/generic/tcpclient.h File Reference	316
9.82 tcpclient.h	317
9.83 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/← Source/driver/bleuart.cpp File Reference	318
9.84 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/← Source/driver/geoid_helper.cpp File Reference	318
9.84.1 Detailed Description	319
9.85 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/← Source/driver/gpx_parse.cpp File Reference	319
9.85.1 Function Documentation	320
9.85.1.1 haversineDistance()	320
9.85.2 Variable Documentation	321
9.85.2.1 EARTH_RADIUS	321
9.86 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/← Source/driver/mqttclient.cpp File Reference	321
9.86.1 Detailed Description	321
9.87 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/← Source/driver/rotation_matrix.cpp File Reference	321
9.87.1 Detailed Description	322
9.87.2 Function Documentation	322
9.87.2.1 createRotationMatrix()	322
9.87.2.2 createRotationMatrixZ()	323
9.87.2.3 multiplyMatrix()	323
9.87.2.4 printVector()	323
9.87.2.5 rotateSensors() [1/3]	323
9.87.2.6 rotateSensors() [2/3]	324
9.87.2.7 rotateSensors() [3/3]	324
9.87.2.8 rotateVector()	325
9.88 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/← Source/driver/serialport.cpp File Reference	325
9.88.1 Detailed Description	325
9.89 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/← Source/driver/serialport_android.cpp File Reference	326
9.89.1 Function Documentation	326
9.89.1.1 getSelf()	326
9.89.1.2 Java_com_hoho_android_usbserial_driver_TestClassTerje_nativeOnConnected() .	327
9.89.1.3 Java_com_hoho_android_usbserial_driver_TestClassTerje_nativeOnSerialBytes() .	327
9.89.1.4 Java_com_hoho_android_usbserial_driver_TestClassTerje_nativeOnSerialError() .	328
9.90 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/← Source/driver/serialport_bluetooth.cpp File Reference	328

9.91 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/	329
Source/driver/wit_c_sdk.c File Reference	329
9.91.1 Macro Definition Documentation	330
9.91.1.1 FuncR	330
9.91.1.2 FuncW	330
9.91.2 Function Documentation	331
9.91.2.1 __CaliSum()	331
9.91.2.2 __CRC16()	331
9.91.2.3 CheckRange()	331
9.91.2.4 CopeWitData()	332
9.91.2.5 posix_delay_ms()	332
9.91.2.6 WitCalRefAngle()	332
9.91.2.7 WitDeInit()	333
9.91.2.8 WitDelayMsRegister()	333
9.91.2.9 WitInit()	333
9.91.2.10 WitReadReg()	333
9.91.2.11 WitRegisterCallBack()	334
9.91.2.12 WitSaveParameter()	334
9.91.2.13 WitSerialDataIn()	334
9.91.2.14 WitSerialWriteRegister()	335
9.91.2.15 WitSetBandwidth()	335
9.91.2.16 WitSetCanBaud()	336
9.91.2.17 WitSetContent()	336
9.91.2.18 WitSetForReset()	336
9.91.2.19 WitSetOutputRate()	337
9.91.2.20 WitSetUartBaud()	337
9.91.2.21 WitStartAccCali()	338
9.91.2.22 WitStartMagCali()	338
9.91.2.23 WitStopAccCali()	338
9.91.2.24 WitStopMagCali()	339
9.91.2.25 WitWriteReg()	339
9.91.3 Variable Documentation	340
9.91.3.1 __ auchCRCHi	340
9.91.3.2 __ auchCRCLO	341
9.91.3.3 p_WitDelaymsFunc	341
9.91.3.4 p_WitRegUpdateCbFunc	341
9.91.3.5 p_WitSerialWriteFunc	341
9.91.3.6 s_ucAddr	341
9.91.3.7 s_ucWitDataBuff	341
9.91.3.8 s_uiProtocol	342
9.91.3.9 s_uiReadRegIndex	342
9.91.3.10 s_uiWitDataCnt	342

9.91.3.11 sReg	342
9.92 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/← Source/generic/characteristicinfo.cpp File Reference	342
9.93 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/← Source/generic/chatclient.cpp File Reference	343
9.94 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/← Source/generic/chatserver.cpp File Reference	343
9.94.1 Function Documentation	344
9.94.1.1 serviceUuid()	344
9.95 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/← Source/generic/device.cpp File Reference	344
9.95.1 Macro Definition Documentation	345
9.95.1.1 key	345
9.95.2 Variable Documentation	345
9.95.2.1 channels	345
9.96 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/← Source/generic/deviceinfo.cpp File Reference	345
9.97 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/← Source/generic/ins_driver.cpp File Reference	346
9.97.1 Detailed Description	347
9.97.2 Typedef Documentation	347
9.97.2.1 RxCallbackINS	347
9.97.3 Function Documentation	347
9.97.3.1 AutoScanSensor()	347
9.97.3.2 AutoSetBaud()	348
9.97.3.3 CopeSensorData()	348
9.97.3.4 INS_driver()	348
9.97.3.5 SensorUartSend()	349
9.97.3.6 setINSRxCallback()	350
9.97.4 Variable Documentation	350
9.97.4.1 callback_	350
9.97.4.2 handle	350
9.97.4.3 iAddress	350
9.97.4.4 iComPort	350
9.97.4.5 s_cDataUpdate	351
9.97.4.6 serialPortb	351
9.97.4.7 serialPorts	351
9.98 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/← Source/generic/lockhelper.cpp File Reference	351
9.99 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/main.cpp File Reference	352
9.99.1 Function Documentation	352
9.99.1.1 main()	352
9.100 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/← Source/generic/main.cpp File Reference	352

9.100.1 Detailed Description	353
9.100.2 Function Documentation	353
9.100.2.1 main()	353
9.101 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/← MainWindow.cpp File Reference	353
9.102 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/← Source/generic/mainwindow.cpp File Reference	354
9.102.1 Detailed Description	355
9.102.2 Macro Definition Documentation	355
9.102.2.1 filterlength	355
9.102.2.2 GFILTER	355
9.102.2.3 headfilterlength	355
9.102.2.4 rotfilterlength	355
9.102.2.5 SCALE	356
9.102.2.6 varfilterlength	356
9.103 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/← Source/generic/MicroQiskitCpp.cpp File Reference	356
9.103.1 Function Documentation	356
9.103.1.1 Qiskit()	356
9.104 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/← Source/generic/multicastlistner.cpp File Reference	357
9.105 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/← Source/generic/MyDevice.cpp File Reference	358
9.106 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/← Source/generic/mytcpsocket.cpp File Reference	358
9.106.1 Detailed Description	359
9.106.2 Macro Definition Documentation	359
9.106.2.1 UDP	359
9.106.3 Function Documentation	360
9.106.3.1 AutoScanSensor()	360
9.106.3.2 AutoSetBaud()	360
9.106.3.3 INS_driver()	361
9.106.3.4 join32()	361
9.106.3.5 nmea_ddmm_to_deg()	362
9.107 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/← Source/generic/remoteselector.cpp File Reference	362
9.108 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/← Source/generic/serviceinfo.cpp File Reference	363
9.109 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/← Source/generic/ssdp.cpp File Reference	363
9.109.1 Detailed Description	364
9.110 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/← Source/generic/tcpclient.cpp File Reference	364
9.110.1 Detailed Description	365

Chapter 1

Directory Hierarchy

1.1 Directories

driver	15
bleuart.h	202
geoid_helper.h	207
gpx_parse.h	208
mqttclient.h	209
rotation_matrix.h	212
serialport.h	218
serialport_bluetooth.h	219
wit_c_sdk.h	221
driver	16
bleuart.cpp	318
geoid_helper.cpp	318
gpx_parse.cpp	319
mqttclient.cpp	321
rotation_matrix.cpp	321
serialport.cpp	325
serialport_android.cpp	326
serialport_bluetooth.cpp	328
wit_c_sdk.c	329
example	16
LayoutSquare.cpp	175
LayoutSquare.h	176
main.cpp	352
MainWindow.cpp	353
MainWindow.h	247
WidgetAI.cpp	178
WidgetAI.h	178
WidgetALT.cpp	180
WidgetALT.h	181
WidgetASI.cpp	183
WidgetASI.h	183
WidgetEADI.cpp	185
WidgetEADI.h	186
WidgetEHSI.cpp	189
WidgetEHSI.h	189
WidgetHI.cpp	192
WidgetHI.h	192

WidgetSix.cpp	194
WidgetSix.h	195
WidgetTC.cpp	197
WidgetTC.h	198
WidgetVSI.cpp	200
WidgetVSI.h	200
generic	17
characteristicinfo.h	236
chatclient.h	237
chatserver.h	239
device.h	241
deviceinfo.h	243
lockhelper.h	245
mainwindow.h	249
MicroQiskitCpp.h	258
multicastlistner.h	266
MyDevice.h	268
myNativeWrapperFunctions.h	269
mytcpsocket.h	270
REG.h	277
remoteselector.h	310
serviceinfo.h	312
ssdp.h	313
tcpclient.h	316
generic	18
characteristicinfo.cpp	342
chatclient.cpp	343
chatserver.cpp	343
device.cpp	344
deviceinfo.cpp	345
ins_driver.cpp	346
lockhelper.cpp	351
main.cpp	352
mainwindow.cpp	354
MicroQiskitCpp.cpp	356
multicastlistner.cpp	357
MyDevice.cpp	358
mytcpsocket.cpp	358
remoteselector.cpp	362
serviceinfo.cpp	363
ssdp.cpp	363
tcpclient.cpp	364
Header	19
driver	15
bleuart.h	202
geoid_helper.h	207
gpx_parse.h	208
mqttclient.h	209
rotation_matrix.h	212
serialport.h	218
serialport_bluetooth.h	219
wit_c_sdk.h	221
generic	17
characteristicinfo.h	236
chatclient.h	237
chatserver.h	239
device.h	241

deviceinfo.h	243
lockhelper.h	245
mainwindow.h	249
MicroQiskitCpp.h	258
multicastlistner.h	266
MyDevice.h	268
myNativeWrapperFunctions.h	269
mytcpsocket.h	270
REG.h	277
remoteselector.h	310
serviceinfo.h	312
ssdp.h	313
tcpclient.h	316
Source	20
driver	16
bleuart.cpp	318
geoid_helper.cpp	318
gpx_parse.cpp	319
mqttclient.cpp	321
rotation_matrix.cpp	321
serialport.cpp	325
serialport_android.cpp	326
serialport_bluetooth.cpp	328
wit_c_sdk.c	329
generic	18
characteristicinfo.cpp	342
chatclient.cpp	343
chatserver.cpp	343
device.cpp	344
deviceinfo.cpp	345
ins_driver.cpp	346
lockhelper.cpp	351
main.cpp	352
mainwindow.cpp	354
MicroQiskitCpp.cpp	356
multicastlistner.cpp	357
MyDevice.cpp	358
mytcpsocket.cpp	358
remoteselector.cpp	362
serviceinfo.cpp	363
ssdp.cpp	363
tcpclient.cpp	364

Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Ui	21
--------------------	-------	----

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

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

AltimeterData	23
MyTcpSocket::Callbacks	31
GeoidHelper	64
GeoidResult	65
GpxParser	66
mqtt::iaction_listener	
ConnListener	55
GeoidHelper::Impl	67
KeepAwakeHelper	67
LFD	74
MqttClient	92
MyDevice	98
PortEntry	126
QDialog	
NoButtonMessageBox	124
RemoteSelector	135
QLayout	
LayoutSquare	68
QMainWindow	
MainWindow	74
MainWindow	74
QObject	
BTComQt	27
BleUart	24
CharacteristicInfo	32
ChatClient	34
ChatServer	38
ComBt	42
ComQt	48
Device	57
DeviceInfo	62
MulticastListener	96
MyTcpSocket	99
ServiceInfo	138
SsdpDiscoverer	143

TcpClient	146
QuantumCircuit	127
QWidget	
WidgetAI	149
WidgetALT	151
WidgetASI	154
WidgetEADI	156
WidgetEHSI	162
WidgetHI	165
WidgetSix	167
WidgetTC	169
WidgetVSI	172
SensorService	137
Simulator	140
TrackPoint	148

Chapter 4

Data Structure Index

4.1 Data Structures

Here are the data structures with brief descriptions:

AltimeterData	23
BleUart	24
BTComQt	27
MyTcpSocket::Callbacks	31
CharacteristicInfo	32
ChatClient [declaration]	34
ChatServer [declaration]	38
ComBt	42
ComQt Thin wrapper around QSerialPort used by MyTcpSocket	48
ConnListener Simple connection listener for connect() operations	55
Device	57
DeviceInfo	62
GeoidHelper	64
GeoidResult	65
GpxParser	66
GeoidHelper::Impl	67
KeepAwakeHelper	67
LayoutSquare This class provides constant aspect ration (square) layout for a single widget	68
LFD	74
MainWindow Main UI window for the Glasscockpit 200-UAV	74
MqttClient Wrapper around mqtt::async_client for "v3-style" MQTT usage	92
MulticastListener	96
MyDevice	98
MyTcpSocket Handles communication with:	99
NoButtonMessageBox Small frameless dialog for transient status messages	124
PortEntry Convenience record for describing a serial port (used on macOS)	126

QuantumCircuit	127
RemoteSelector	135
SensorService	137
ServiceInfo	138
Simulator	140
SsdpDiscoverer	143
TcpClient	146
TrackPoint	148
WidgetAI	149
WidgetALT	151
WidgetASI	154
WidgetEADI	156
WidgetEHSI	162
WidgetHI	165
WidgetSix	167
WidgetTC	169
WidgetVSI	172

Chapter 5

File Index

5.1 File List

Here is a list of all files with brief descriptions:

```
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/LayoutSquare.cpp  
    175  
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/LayoutSquare.h  
    176  
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/main.cpp  
    352  
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/MainWindow.cpp  
    353  
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/MainWindow.h  
    247  
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/WidgetAI.cpp  
    178  
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/WidgetAI.h  
    178  
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/WidgetALT.cpp  
    180  
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/WidgetALT.h  
    181  
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/WidgetASI.cpp  
    183  
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/WidgetASI.h  
    183  
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/WidgetEADI.cpp  
    185  
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/WidgetEADI.h  
    186  
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/WidgetEHSI.cpp  
    189  
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/WidgetEHSI.h  
    189  
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/WidgetHI.cpp  
    192  
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/WidgetHI.h  
    192  
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/WidgetSix.cpp  
    194
```

/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/ WidgetSix.h	195
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/ WidgetTC.cpp	197
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/ WidgetTC.h	198
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/ WidgetVSI.cpp	200
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/ WidgetVSI.h	200
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header/driver/ bleuart.h	202
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header/driver/ geoid_helper.h	207
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header/driver/ gpx_parse.h	208
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header/driver/ mqttclient.h	209
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header/driver/ rotation_matrix.h	212
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header/driver/ serialport.h	218
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header/driver/ serialport_bluetooth.h	219
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header/driver/ wit_c_sdk.h	221
This file provides all Configure sensor function	
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header/generic/ characteristicinfo.h	236
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header/generic/ chatclient.h	237
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header/generic/ chatserver.h	239
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header/generic/ device.h	241
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header/generic/ deviceinfo.h	243
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header/generic/ lockhelper.h	245
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header/generic/ mainwindow.h	249
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header/generic/ MicroQiskitCpp.h	258
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header/generic/ multicastlistner.h	266
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header/generic/ MyDevice.h	268
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header/generic/ myNativeWrapperFun.h	269
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header/generic/ mytcpsocket.h	270
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header/generic/ REG.h	277
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header/generic/ remoteselector.h	310
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header/generic/ serviceinfo.h	312
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header/generic/ ssdp.h	313
SSDP (Simple Service Discovery Protocol) discovery support	

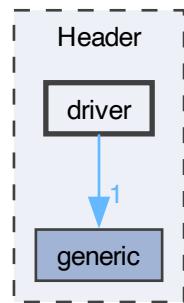
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header/generic/	tcpclient.h
	316
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/driver/	bleuart.cpp
	318
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/driver/	geoid_helper.cpp
	Geoid helper implementation 318
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/driver/	gpx_parse.cpp
	319
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/driver/	mqttclient.cpp
	Implementation of MqttClient 321
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/driver/	rotation_matrix.cpp
	Implementation of createRotationMatrix 321
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/driver/	serialport.cpp
	Implementation of ComQt 325
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/driver/	serialport_android.cpp
	326
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/driver/	serialport_bluetooth.cpp
	328
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/driver/	wit_c_sdk.c
	329
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/generic/	characteristicinfo.cpp
	342
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/generic/	chatclient.cpp
	343
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/generic/	chatserver.cpp
	343
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/generic/	device.cpp
	344
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/generic/	deviceinfo.cpp
	345
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/generic/	ins_driver.cpp
	Implementation of INS_driver 346
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/generic/	lockhelper.cpp
	351
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/generic/	main.cpp
	Implementation of main 352
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/generic/	mainwindow.cpp
	Implementation of MainWindow 354
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/generic/	MicroQiskitCpp.cpp
	356
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/generic/	multicastlistner.cpp
	357
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/generic/	MyDevice.cpp
	358
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/generic/	mytcpsocket.cpp
	Implementation of MyTcpSocket 358
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/generic/	remoteselector.cpp
	362
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/generic/	serviceinfo.cpp
	363
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/generic/	ssdp.cpp
	Implementation of SSDP discovery support 363
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/generic/	tcpclient.cpp
	Implementation of TcpClient 364

Chapter 6

Directory Documentation

6.1 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_← 11_25_Serial_IMU_only/Header/driver Directory Reference

Directory dependency graph for driver:



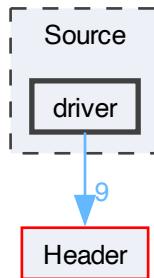
Files

- file [bleuart.h](#)
- file [geoid_helper.h](#)
- file [gpx_parse.h](#)
- file [mqttclient.h](#)
- file [rotation_matrix.h](#)
- file [serialport.h](#)
- file [serialport_bluetooth.h](#)
- file [wit_c_sdk.h](#)

This file provides all Configure sensor function.

6.2 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_← 11_25_Serial_IMU_only/Source/driver Directory Reference

Directory dependency graph for driver:



Files

- file [bleuart.cpp](#)
- file [geoid_helper.cpp](#)
Geoid helper implementation.
- file [gpx_parse.cpp](#)
- file [mqttclient.cpp](#)
Implementation of MqttClient.
- file [rotation_matrix.cpp](#)
Implementation of createRotationMatrix.
- file [serialport.cpp](#)
Implementation of ComQt.
- file [serialport_android.cpp](#)
- file [serialport_bluetooth.cpp](#)
- file [wit_c_sdk.c](#)

6.3 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_← 11_25_Serial_IMU_only/example Directory Reference

Files

- file [LayoutSquare.cpp](#)
- file [LayoutSquare.h](#)
- file [main.cpp](#)
- file [MainWindow.cpp](#)
- file [MainWindow.h](#)
- file [WidgetAI.cpp](#)
- file [WidgetAI.h](#)

6.4

/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header/generic

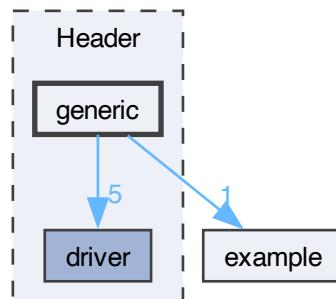
Directory Reference

-
- file [WidgetALT.cpp](#)
 - file [WidgetALT.h](#)
 - file [WidgetASI.cpp](#)
 - file [WidgetASI.h](#)
 - file [WidgetEADI.cpp](#)
 - file [WidgetEADI.h](#)
 - file [WidgetEHSI.cpp](#)
 - file [WidgetEHSI.h](#)
 - file [WidgetHI.cpp](#)
 - file [WidgetHI.h](#)
 - file [WidgetSix.cpp](#)
 - file [WidgetSix.h](#)
 - file [WidgetTC.cpp](#)
 - file [WidgetTC.h](#)
 - file [WidgetVSI.cpp](#)
 - file [WidgetVSI.h](#)

17

6.4 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header/generic Directory Reference

Directory dependency graph for generic:



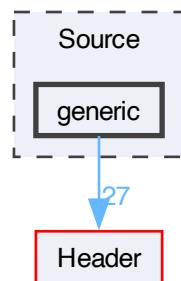
Files

- file [characteristicinfo.h](#)
- file [chatclient.h](#)
- file [chatserver.h](#)
- file [device.h](#)
- file [deviceinfo.h](#)
- file [lockhelper.h](#)
- file [mainwindow.h](#)
- file [MicroQiskitCpp.h](#)
- file [multicastlistner.h](#)

- file [MyDevice.h](#)
 - file [myNativeWrapperFunctions.h](#)
 - file [mytcpsocket.h](#)
 - file [REG.h](#)
 - file [remoteselector.h](#)
 - file [serviceinfo.h](#)
 - file [ssdp.h](#)
- SSDP (Simple Service Discovery Protocol) discovery support.*
- file [tcpclient.h](#)

6.5 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_← 11_25_Serial_IMU_only/Source/generic Directory Reference

Directory dependency graph for generic:



Files

- file [characteristicinfo.cpp](#)
- file [chatclient.cpp](#)
- file [chatserver.cpp](#)
- file [device.cpp](#)
- file [deviceinfo.cpp](#)
- file [ins_driver.cpp](#)

Implementation of INS_driver.
- file [lockhelper.cpp](#)
- file [main.cpp](#)

Implementation of main.
- file [mainwindow.cpp](#)

Implementation of MainWindow.
- file [MicroQiskitCpp.cpp](#)
- file [multicastlistner.cpp](#)
- file [MyDevice.cpp](#)
- file [mytcpsocket.cpp](#)

Implementation of MyTcpSocket.

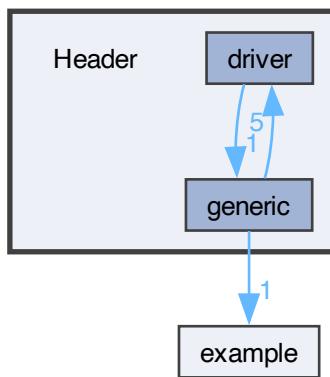
- file [remoteselector.cpp](#)
- file [serviceinfo.cpp](#)
- file [ssdp.cpp](#)

Implementation of SSDP discovery support.
- file [tcpclient.cpp](#)

Implementation of TcpClient.

6.6 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header Directory Reference

Directory dependency graph for Header:

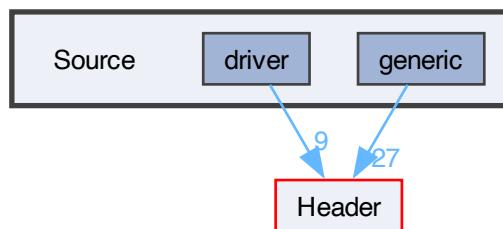


Directories

- directory [driver](#)
- directory [generic](#)

6.7 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_← 11_25_Serial_IMU_only/Source Directory Reference

Directory dependency graph for Source:



Directories

- directory [driver](#)
- directory [generic](#)

Chapter 7

Namespace Documentation

7.1 Ui Namespace Reference

Chapter 8

Data Structure Documentation

8.1 AltimeterData Struct Reference

```
#include <mytcpsocket.h>
```

Data Fields

- float `pressure`
- float `temperature`
- float `relative`
- float `altitude`

8.1.1 Field Documentation

8.1.1.1 `altitude`

```
float AltimeterData::altitude
```

8.1.1.2 `pressure`

```
float AltimeterData::pressure
```

8.1.1.3 `relative`

```
float AltimeterData::relative
```

8.1.1.4 `temperature`

```
float AltimeterData::temperature
```

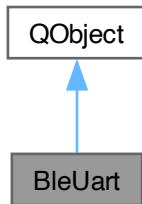
The documentation for this struct was generated from the following file:

- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header/generic/mytcpsocket.h

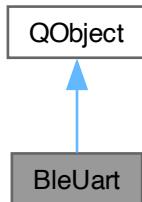
8.2 BleUart Class Reference

```
#include <bleuart.h>
```

Inheritance diagram for BleUart:



Collaboration diagram for BleUart:



Public Types

- using `RxCallback = void(*)(void *handler, const char* data, uint32_t length)`

Public Member Functions

- void `startScan` (int timeoutMs=5000, int rssiMin=-95)
- void `connectTo` (const QBluetoothDeviceInfo &dev)
- void `writeBytes` (const QByteArray &data)
- void `bytesReceived` (QByteArray)
- void `ready` ()
- void `disconnected` ()
- void `dataReceived` (QByteArray)
- void `setRxCallback` (`RxCallback` cb)

Data Fields

- bool `scancomplete` = false

8.2.1 Member Typedef Documentation

8.2.1.1 RxCallback

```
using BleUart::RxCallback = void(*)(void *handler, const char* data, uint32_t length)
```

8.2.2 Member Function Documentation

8.2.2.1 bytesReceived()

```
void BleUart::bytesReceived ( QByteArray )
```

Here is the caller graph for this function:



8.2.2.2 connectTo()

```
void BleUart::connectTo ( const QBluetoothDeviceInfo & dev) [inline]
```

8.2.2.3 dataReceived()

```
void BleUart::dataReceived ( QByteArray )
```

Here is the call graph for this function:



8.2.2.4 disconnected()

```
void BleUart::disconnected ()
```

Here is the caller graph for this function:



8.2.2.5 ready()

```
void BleUart::ready ()
```

Here is the caller graph for this function:



8.2.2.6 setRxCallback()

```
void BleUart::setRxCallback (
    RxCallback cb) [inline]
```

8.2.2.7 startScan()

```
void BleUart::startScan (
    int timeoutMs = 5000,
    int rssiMin = -95) [inline]
```

8.2.2.8 writeBytes()

```
void BleUart::writeBytes (
    const QByteArray & data) [inline]
```

8.2.3 Field Documentation

8.2.3.1 scancomplete

```
bool BleUart::scancomplete = false
```

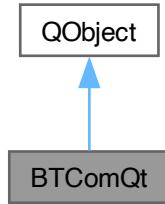
The documentation for this class was generated from the following file:

- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header/driver/[bleuart.h](#)

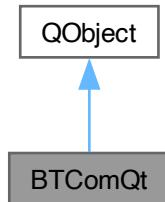
8.3 BTComQt Class Reference

```
#include <serialport_bluetooth.h>
```

Inheritance diagram for BTComQt:



Collaboration diagram for BTComQt:



Public Types

- using [RxCallback](#) = void(*)(void *handler, const char* data, uint32_t length)

Public Member Functions

- `BTComQt (QObject *parent=nullptr)`
- `~BTComQt ()`
- `bool open (const QString &portName, qint32 baudrate)`
- `void close ()`
- `bool send (const QByteArray &data)`
- `bool send (const char *data, unsigned short len)`
- `void setRxCallback (RxCallback cb)`
- `void dataReceived (const QByteArray &data)`

Data Fields

- `QJniObject * imuJavaObject = nullptr`
- `QSerialPort * serial_ = nullptr`
- `QTimer * timerAndroid`

8.3.1 Member Typedef Documentation

8.3.1.1 RxCallback

```
using BTComQt::RxCallback = void(*)(void *handler, const char* data, uint32_t length)
```

8.3.2 Constructor & Destructor Documentation

8.3.2.1 BTComQt()

```
BTComQt::BTComQt (
    QObject * parent = nullptr) [explicit]
```

8.3.2.2 ~BTComQt()

```
BTComQt::~BTComQt ()
```

Here is the call graph for this function:

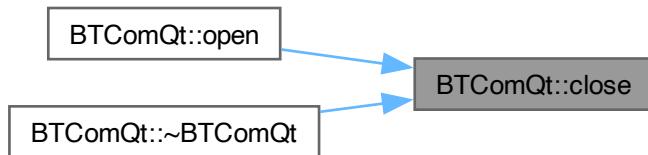


8.3.3 Member Function Documentation

8.3.3.1 close()

```
void BTComQt::close ()
```

Here is the caller graph for this function:



8.3.3.2 dataReceived()

```
void BTComQt::dataReceived (
    const QByteArray & data)
```

8.3.3.3 open()

```
bool BTComQt::open (
    const QString & portName,
    qint32 baudrate)
```

Here is the call graph for this function:



8.3.3.4 send() [1/2]

```
bool BTComQt::send (
    const char * data,
    unsigned short len)
```

Here is the call graph for this function:



8.3.3.5 send() [2/2]

```
bool BTComQt::send (
    const QByteArray & data)
```

Here is the caller graph for this function:



8.3.3.6 setRxCallback()

```
void BTComQt::setRxCallback (
    RxCallback cb) [inline]
```

8.3.4 Field Documentation

8.3.4.1 imuJavaObject

```
QJniObject* BTComQt::imuJavaObject = nullptr
```

8.3.4.2 serial_

```
QSerialPort* BTComQt::serial_ = nullptr
```

8.3.4.3 timerAndroid

```
QTimer* BTComQt::timerAndroid
```

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

- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header/driver/[serialport_bluetooth.h](#)
- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/driver/[serialport_bluetooth.h](#)

8.4 MyTcpSocket::Callbacks Struct Reference

```
#include <mytcpsocket.h>
```

Data Fields

- void * [classPtr](#)
- void(* [callback](#))(void *)

8.4.1 Field Documentation

8.4.1.1 [callback](#)

```
void(* MyTcpSocket::Callbacks::callback) (void *)
```

8.4.1.2 [classPtr](#)

```
void* MyTcpSocket::Callbacks::classPtr
```

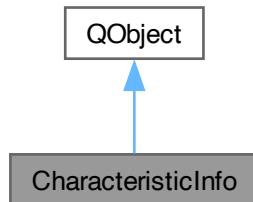
The documentation for this struct was generated from the following file:

- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header/generic/[mytcpsocket.h](#)

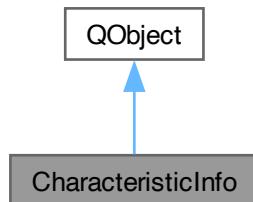
8.5 CharacteristicInfo Class Reference

```
#include <characteristicinfo.h>
```

Inheritance diagram for CharacteristicInfo:



Collaboration diagram for CharacteristicInfo:



Public Member Functions

- `CharacteristicInfo ()=default`
- `CharacteristicInfo (const QLowEnergyCharacteristic &characteristic)`
- `void setCharacteristic (const QLowEnergyCharacteristic &characteristic)`
- `QString getName () const`
- `QString getUuid () const`
- `QString getValue () const`
- `QString getPermission () const`
- `QLowEnergyCharacteristic getCharacteristic () const`
- `void characteristicChanged ()`

8.5.1 Constructor & Destructor Documentation

8.5.1.1 CharacteristicInfo() [1/2]

```
CharacteristicInfo::CharacteristicInfo () [default]
```

8.5.1.2 CharacteristicInfo() [2/2]

```
CharacteristicInfo::CharacteristicInfo (
    const QLowEnergyCharacteristic & characteristic)
```

8.5.2 Member Function Documentation

8.5.2.1 characteristicChanged()

```
void CharacteristicInfo::characteristicChanged ()
```

Here is the caller graph for this function:



8.5.2.2 getCharacteristic()

```
QLowEnergyCharacteristic CharacteristicInfo::getCharacteristic () const
```

8.5.2.3 getName()

```
QString CharacteristicInfo::getName () const
```

[les-get-descriptors]

[les-get-descriptors]

8.5.2.4 getPermission()

```
QString CharacteristicInfo::getPermission () const
```

8.5.2.5 getUuid()

```
QString CharacteristicInfo::getUuid () const
```

8.5.2.6 `getValue()`

```
QString CharacteristicInfo::getValue () const
```

8.5.2.7 `setCharacteristic()`

```
void CharacteristicInfo::setCharacteristic (
    const QLowEnergyCharacteristic & characteristic)
```

Here is the call graph for this function:



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

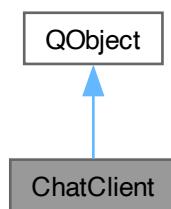
- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header/generic/[characteristicinfo.h](#)
- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/generic/[characteristicinfo.cpp](#)

8.6 ChatClient Class Reference

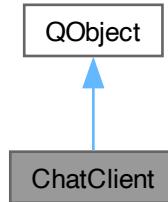
[declaration]

```
#include <chatclient.h>
```

Inheritance diagram for ChatClient:



Collaboration diagram for ChatClient:



Public Member Functions

- `ChatClient (QObject *parent=nullptr)`
- `~ChatClient ()`
- `void startClient (const QBluetoothServiceInfo &remoteService)
[startClient]`
- `void stopClient ()
[startClient]`
- `void sendMessage (const QString &message)
[readSocket]`
- `void messageReceived (const QString &sender, const QString &message)`
- `void connected (const QString &name)`
- `void disconnected ()`
- `void socketErrorOccurred (const QString &errorString)`

8.6.1 Detailed Description

[declaration]

8.6.2 Constructor & Destructor Documentation

8.6.2.1 ChatClient()

```
ChatClient::ChatClient ( QObject * parent = nullptr) [explicit]
```

8.6.2.2 ~ChatClient()

```
ChatClient::~ChatClient ()
```

Here is the call graph for this function:



8.6.3 Member Function Documentation

8.6.3.1 connected()

```
void ChatClient::connected (
    const QString & name)
```

Here is the caller graph for this function:



8.6.3.2 disconnected()

```
void ChatClient::disconnected ()
```

Here is the caller graph for this function:



8.6.3.3 messageReceived()

```
void ChatClient::messageReceived (
    const QString & sender,
    const QString & message)
```

8.6.3.4 sendMessage()

```
void ChatClient::sendMessage (
    const QString & message)
```

[readSocket]

[sendMessage]

8.6.3.5 socketErrorOccurred()

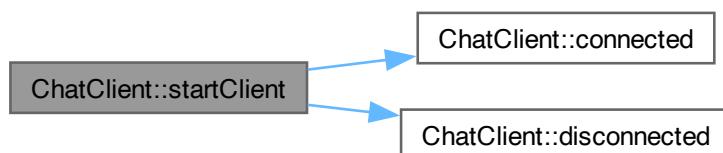
```
void ChatClient::socketErrorOccurred (
    const QString & errorString)
```

8.6.3.6 startClient()

```
void ChatClient::startClient (
    const QBluetoothServiceInfo & remoteService)
```

[startClient]

Here is the call graph for this function:



8.6.3.7 stopClient()

```
void ChatClient::stopClient ()
```

[startClient]

[stopClient] Here is the caller graph for this function:



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

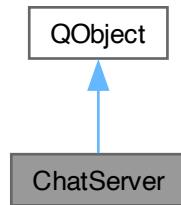
- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header/generic/[chatclient.h](#)
- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/generic/[chatclient.cpp](#)

8.7 ChatServer Class Reference

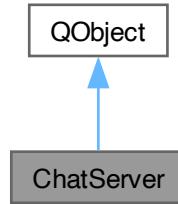
[declaration]

```
#include <chatserver.h>
```

Inheritance diagram for ChatServer:



Collaboration diagram for ChatServer:



Public Member Functions

- `ChatServer (QObject *parent=nullptr)`
[Service UUID]
- `~ChatServer ()`
- `void startServer (const QBluetoothAddress &localAdapter=QBluetoothAddress())`
- `void stopServer ()`
[stopServer]
- `void sendMessage (const QString &message)`
[stopServer]
- `void messageReceived (const QString &sender, const QString &message)`
- `void clientConnected (const QString &name)`
- `void clientDisconnected (const QString &name)`

8.7.1 Detailed Description

[declaration]

8.7.2 Constructor & Destructor Documentation

8.7.2.1 ChatServer()

```
ChatServer::ChatServer (
    QObject * parent = nullptr) [explicit]
```

[Service UUID]

8.7.2.2 ~ChatServer()

```
ChatServer::~ChatServer ()
```

Here is the call graph for this function:



8.7.3 Member Function Documentation

8.7.3.1 clientConnected()

```
void ChatServer::clientConnected (
    const QString & name)
```

Here is the caller graph for this function:



8.7.3.2 clientDisconnected()

```
void ChatServer::clientDisconnected (
    const QString & name)
```

8.7.3.3 messageReceived()

```
void ChatServer::messageReceived (
    const QString & sender,
    const QString & message)
```

8.7.3.4 sendMessage()

```
void ChatServer::sendMessage (
    const QString & message)
```

[stopServer]

[sendMessage]

8.7.3.5 startServer()

```
void ChatServer::startServer (
    const QBluetoothAddress & localAdapter = QBluetoothAddress())
```

[Create the server]

[Create the server]

[Service name, description and provider]

[Service name, description and provider]

[Service UUID set]

[Service UUID set]

[Service Discoverability]

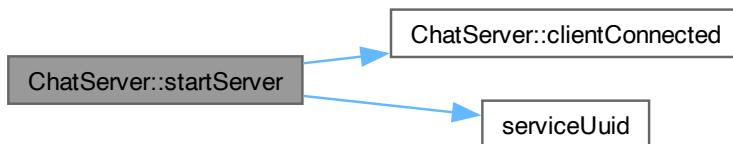
[Service Discoverability]

[Protocol descriptor list]

[Protocol descriptor list]

[Register service]

[Register service]Here is the call graph for this function:



8.7.3.6 stopServer()

```
void ChatServer::stopServer ()
```

[stopServer]

Here is the caller graph for this function:



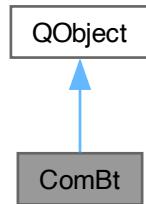
The documentation for this class was generated from the following files:

- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header/generic/[chatserver.h](#)
- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/generic/[chatserver.cpp](#)

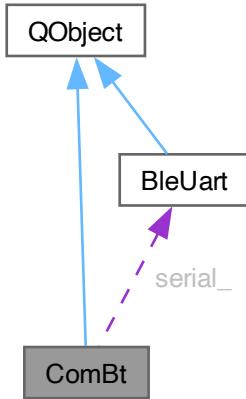
8.8 ComBt Class Reference

```
#include <bleuart.h>
```

Inheritance diagram for ComBt:



Collaboration diagram for ComBt:



Public Types

- using `RxCallback = void(*)(void* handler, const char* data, uint32_t length)`

Public Member Functions

- `ComBt (QObject *parent=nullptr)`
- `~ComBt ()`
- `bool open (const QString &portName=QString(), qint32 baudrate=0)`
- `void close ()`
- `bool setBaudrate (qint32)`
- `bool send (const QByteArray &data)`
- `bool send (const char *data, unsigned short len)`
- `void setRxCallback (RxCallback cb)`
- `void connectionChanged (bool connected)`
- `void dataReceived (const QByteArray &data)`
- `void errorReceived (const QString &message)`

Data Fields

- `BleUart * serial_ = nullptr`
- `bool scanDone = false`
- `const QString status`
- `std::function< void(void *, const char *, uint32_t)> callback_`
- `QObject * parent`

8.8.1 Member Typedef Documentation

8.8.1.1 RxCallback

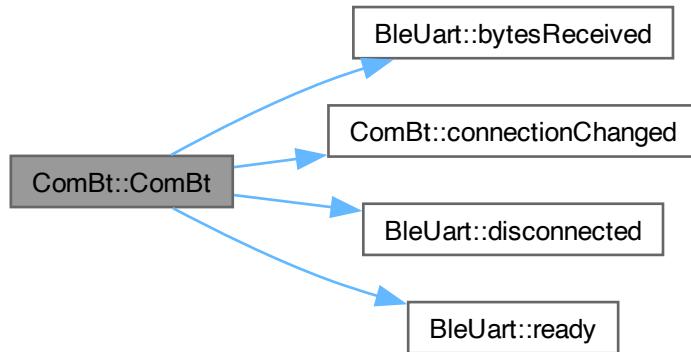
```
using ComBt::RxCallback = void(*) (void* handler, const char* data, uint32_t length)
```

8.8.2 Constructor & Destructor Documentation

8.8.2.1 ComBt()

```
ComBt::ComBt (
    QObject * parent = nullptr) [explicit]
```

Here is the call graph for this function:



Here is the caller graph for this function:



8.8.2.2 ~ComBt()

```
ComBt::~ComBt ()
```

8.8.3 Member Function Documentation

8.8.3.1 close()

```
void ComBt::close ()
```

8.8.3.2 connectionChanged()

```
void ComBt::connectionChanged (bool connected)
```

Here is the caller graph for this function:



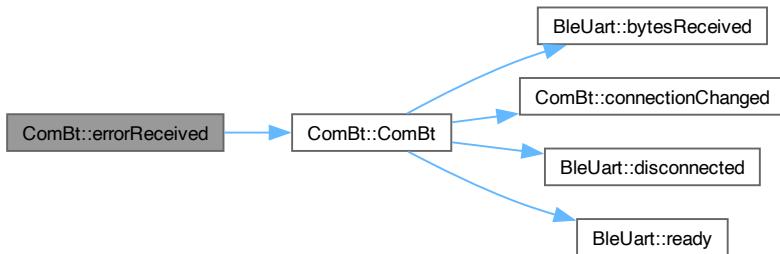
8.8.3.3 dataReceived()

```
void ComBt::dataReceived (const QByteArray & data)
```

8.8.3.4 errorReceived()

```
void ComBt::errorReceived (const QString & message)
```

Here is the call graph for this function:



8.8.3.5 open()

```
bool ComBt::open (
    const QString & portName = QString(),
    qint32 baudrate = 0)
```

8.8.3.6 send() [1/2]

```
bool ComBt::send (
    const char * data,
    unsigned short len) [inline]
```

Here is the call graph for this function:



Here is the caller graph for this function:



8.8.3.7 send() [2/2]

```
bool ComBt::send (
    const QByteArray & data)
```

8.8.3.8 setBaudrate()

```
bool ComBt::setBaudrate (
    qint32 ) [inline]
```

8.8.3.9 setRxCallback()

```
void ComBt::setRxCallback (
    RxCallback cb) [inline]
```

8.8.4 Field Documentation

8.8.4.1 callback_

```
std::function<void(void *, const char*, uint32_t)> ComBt::callback_
```

8.8.4.2 parent

```
QObject* ComBt::parent
```

8.8.4.3 scanDone

```
bool ComBt::scanDone = false
```

8.8.4.4 serial_

```
BleUart* ComBt::serial_ = nullptr
```

8.8.4.5 status

```
const QString ComBt::status
```

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

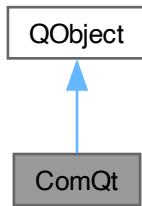
- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header/driver/[bleuart.h](#)
- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/driver/[bleuart.cpp](#)

8.9 ComQt Class Reference

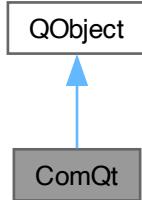
Thin wrapper around QSerialPort used by [MyTcpSocket](#).

```
#include <serialport.h>
```

Inheritance diagram for ComQt:



Collaboration diagram for ComQt:



Public Types

- using **RxCallback** = void(*)(void *handler, const char *data, uint32_t length)
Function pointer type for C-style RX callbacks.

Public Member Functions

- **ComQt** (QObject *parent=nullptr)
- **~ComQt** ()
- bool **open** (const QString &portName, qint32 baudrate)
Open a serial port.
- void **close** ()
Close the serial port if open.

- bool `setBaudrate` (qint32 baudrate)
Change the baudrate of an already-open port.
- bool `send` (const QByteArray &data)
Send a QByteArray over the serial port.
- bool `send` (const char *data, unsigned short len)
Send raw bytes over the serial port.
- bool `isOpen` () const
Convenience wrapper: check if serial port is open.
- void `connectionChanged` (bool connected)
Emitted whenever the connection state changes.
- void `dataReceived` (const QByteArray &data)
Emitted when new data are available on the serial port.
- void `errorReceived` (const QString &message)
Emitted when an error occurs on the serial port.
- void `setRxCallback` (RxCallback cb)
Register a C-style RX callback.

Data Fields

- QSerialPort * `serial_` = nullptr
Underlying QSerialPort instance (nullptr on iOS).
- const QString `status`
Optional human-readable status string (const, set in ctor).
- std::function< void(void *, const char *, uint32_t) > `callback_`
Callback invoked when bytes arrive (if set).
- QObject * `parent` = nullptr
Optional owner pointer (legacy; QObject::parent() is usually preferred).

8.9.1 Detailed Description

Thin wrapper around QSerialPort used by [MyTcpSocket](#).

Responsibilities:

- Open/close a serial port given a port name and baudrate.
- Provide simple `send()` helpers for QByteArray / C-style data.
- Emit Qt signals when data is received, connection changes, or errors occur.
- Optionally forward incoming bytes to a C-style callback (used by C code).

This class is intentionally minimal and is used as the "serial layer" in the rest of the project (transponder, radar, INS).

8.9.2 Member Typedef Documentation

8.9.2.1 RxCallback

```
using ComQt::RxCallback = void(*)(void *handler, const char *data, uint32_t length)
```

Function pointer type for C-style RX callbacks.

Signature must match exactly between typedef and stored std::function.

8.9.3 Constructor & Destructor Documentation

8.9.3.1 ComQt()

```
ComQt::ComQt ( QObject * parent = nullptr) [explicit]
```

8.9.3.2 ~ComQt()

```
ComQt::~ComQt ()
```

Here is the call graph for this function:



8.9.4 Member Function Documentation

8.9.4.1 close()

```
void ComQt::close ()
```

Close the serial port if open.

Here is the caller graph for this function:



8.9.4.2 connectionChanged()

```
void ComQt::connectionChanged (
```

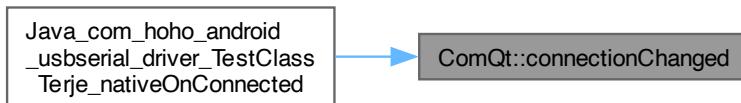
bool	<i>connected</i>
------	------------------

Emitted whenever the connection state changes.

Parameters

<i>connected</i>	true if port is now open, false if it was closed or failed.
------------------	---

Here is the caller graph for this function:



8.9.4.3 dataReceived()

```
void ComQt::dataReceived (
```

const	<code>QByteArray &</code>	<i>data</i>
-------	-------------------------------	-------------

Emitted when new data are available on the serial port.

Here is the caller graph for this function:



8.9.4.4 errorReceived()

```
void ComQt::errorReceived (
    const QString & message)
```

Emitted when an error occurs on the serial port.

Here is the caller graph for this function:



8.9.4.5 isOpen()

```
bool ComQt::isOpen () const [inline]
```

Convenience wrapper: check if serial port is open.

8.9.4.6 open()

```
bool ComQt::open (
    const QString & portName,
    qint32 baudrate)
```

Open a serial port.

Parameters

<i>portName</i>	OS-specific port identifier e.g. "COM4" on Windows, "/dev/tty.usbserial-14130" on macOS, "/dev/ttyUSB0" on Linux, or an Android-specific name.
<i>baudrate</i>	Standard QSerialPort baud rate value (e.g. QSerialPort::Baud9600, QSerialPort::Baud115200).

Returns

true on success, false on failure.

8.9.4.7 send() [1/2]

```
bool ComQt::send (
    const char * data,
    unsigned short len)
```

Send raw bytes over the serial port.

Parameters

<i>data</i>	Pointer to data buffer.
<i>len</i>	Number of bytes to send.

Returns

true if all bytes were queued successfully.

Here is the call graph for this function:



8.9.4.8 send() [2/2]

```
bool ComQt::send (
    const QByteArray & data)
```

Send a QByteArray over the serial port.

Returns

true if all bytes were queued successfully.

Here is the caller graph for this function:



8.9.4.9 setBaudrate()

```
bool ComQt::setBaudrate (
    qint32 baudrate)
```

Change the baudrate of an already-open port.

Returns

true on success, false if port is closed or setting fails.

8.9.4.10 setRxCallback()

```
void ComQt::setRxCallback (
    RxCallback cb) [inline]
```

Register a C-style RX callback.

Only the function pointer is stored here; the user data / handler pointer is provided later when handleReadyRead() is invoked.

8.9.5 Field Documentation

8.9.5.1 callback_

```
std::function<void(void *, const char *, uint32_t)> ComQt::callback_
```

Callback invoked when bytes arrive (if set).

The first argument is a user-provided context pointer (typically a "parent" object that will handle the data).

NOTE: By convention in this project, the context pointer is passed in via [MyTcpSocket](#) when [setRxCallback\(\)](#) is called.

8.9.5.2 parent

```
QObject* ComQt::parent = nullptr
```

Optional owner pointer (legacy; QObject::parent() is usually preferred).

8.9.5.3 serial_

```
QSerialPort* ComQt::serial_ = nullptr
```

Underlying QSerialPort instance (nullptr on iOS).

8.9.5.4 status

```
const QString ComQt::status
```

Optional human-readable status string (const, set in ctor).

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

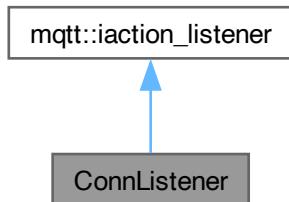
- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header/driver/[serialport.h](#)
- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/driver/[serialport.cpp](#)
- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/driver/[serialport_android.cpp](#)

8.10 ConnListener Class Reference

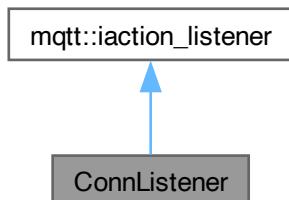
Simple connection listener for connect() operations.

```
#include <mqttclient.h>
```

Inheritance diagram for ConnListener:



Collaboration diagram for ConnListener:



Public Member Functions

- `ConnListener (MqttClient *o)`
- `void on_failure (const mqtt::token &tok) override`
- `void on_success (const mqtt::token &tok) override`

8.10.1 Detailed Description

Simple connection listener for connect() operations.

This uses the legacy mqtt::iaction_listener interface to inform the owning `MqttClient` about connect success/failure.

8.10.2 Constructor & Destructor Documentation

8.10.2.1 ConnListener()

```
ConnListener::ConnListener (
    MqttClient * o) [inline], [explicit]
```

8.10.3 Member Function Documentation

8.10.3.1 on_failure()

```
void ConnListener::on_failure (
    const mqtt::token & tok) [override]
```

8.10.3.2 on_success()

```
void ConnListener::on_success (
    const mqtt::token & tok) [override]
```

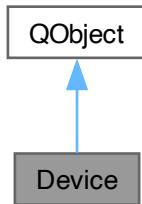
The documentation for this class was generated from the following file:

- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header/driver/mqttclient.h

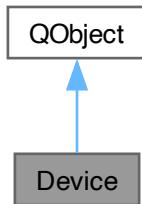
8.11 Device Class Reference

```
#include <device.h>
```

Inheritance diagram for Device:



Collaboration diagram for Device:



Public Member Functions

- [Device \(\)](#)
- [~Device \(\)](#)
- [bool state \(\)](#)
- [void SendData \(QString data, int ch\)](#)
- [void startDeviceDiscovery \(\)](#)
- [void scanServices \(const QString &address\)](#)
- [void connectToService \(const QString &uuid\)](#)
- [void disconnectFromDevice \(\)](#)
- [void setVal \(\)](#)
- [void test \(\)](#)
- [void updateValueSet \(\)](#)
- [void characteristicsUpdated \(\)](#)
- [void updateChanged \(\)](#)
- [void stateChanged \(\)](#)
- [void disconnected \(\)](#)
- [void randomAddressChanged \(\)](#)
- [void updateDisplay \(\)](#)

Data Fields

- QString `m_message`
- QLowEnergyService * `l_service` = nullptr
- QString `key2`
- QString `receivedData`

8.11.1 Constructor & Destructor Documentation

8.11.1.1 `Device()`

```
Device::Device ()
```

[les-devicediscovery-1]

[les-devicediscovery-1]

8.11.1.2 `~Device()`

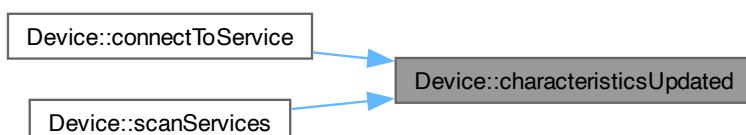
```
Device::~Device ()
```

8.11.2 Member Function Documentation

8.11.2.1 `characteristicsUpdated()`

```
void Device::characteristicsUpdated ()
```

Here is the caller graph for this function:

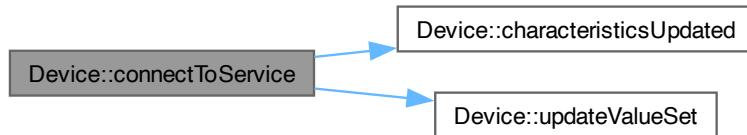


8.11.2.2 connectToService()

```
void Device::connectToService (
    const QString & uuid)
```

[les-service-3]

[les-service-3]Here is the call graph for this function:



8.11.2.3 disconnected()

```
void Device::disconnected ()
```

8.11.2.4 disconnectFromDevice()

```
void Device::disconnectFromDevice ()
```

8.11.2.5 randomAddressChanged()

```
void Device::randomAddressChanged ()
```

8.11.2.6 scanServices()

```
void Device::scanServices (
    const QString & address)
```

[les-controller-1]

[les-controller-1]Here is the call graph for this function:



8.11.2.7 SendData()

```
void Device::SendData (
    QString data,
    int ch)
```

Here is the caller graph for this function:



8.11.2.8 setVal()

```
void Device::setVal ()
```

8.11.2.9 startDeviceDiscovery()

```
void Device::startDeviceDiscovery ()
```

[les-devicediscovery-2]

[les-devicediscovery-2]Here is the call graph for this function:



8.11.2.10 state()

```
bool Device::state ()
```

8.11.2.11 stateChanged()

```
void Device::stateChanged ()
```

Here is the caller graph for this function:

**8.11.2.12 test()**

```
void Device::test ()
```

Here is the call graph for this function:

**8.11.2.13 updateChanged()**

```
void Device::updateChanged ()
```

8.11.2.14 updateDisplay()

```
void Device::updateDisplay ()
```

8.11.2.15 updateValueSet()

```
void Device::updateValueSet ()
```

Here is the caller graph for this function:



8.11.3 Field Documentation

8.11.3.1 key2

```
QString Device::key2
```

8.11.3.2 l_service

```
QLowEnergyService* Device::l_service = nullptr
```

8.11.3.3 m_message

```
QString Device::m_message
```

8.11.3.4 receivedData

```
QString Device::receivedData
```

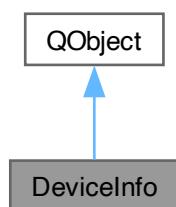
The documentation for this class was generated from the following files:

- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header/generic/[device.h](#)
- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/generic/[device.cpp](#)

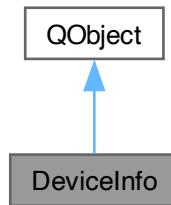
8.12 DevicelInfo Class Reference

```
#include <deviceinfo.h>
```

Inheritance diagram for DevicelInfo:



Collaboration diagram for DeviceInfo:



Public Member Functions

- `DeviceInfo ()=default`
- `DeviceInfo (const QBluetoothDeviceInfo &d)`
- `QString getAddress () const`
- `QString getName () const`
- `QBluetoothDeviceInfo getDevice ()`
- `void setDevice (const QBluetoothDeviceInfo &dev)`
- `void deviceChanged ()`

Properties

- `QString deviceName`

8.12.1 Constructor & Destructor Documentation

8.12.1.1 `DeviceInfo()` [1/2]

```
DeviceInfo::DeviceInfo () [default]
```

8.12.1.2 `DeviceInfo()` [2/2]

```
DeviceInfo::DeviceInfo (
    const QBluetoothDeviceInfo & d)
```

8.12.2 Member Function Documentation

8.12.2.1 `deviceChanged()`

```
void DeviceInfo::deviceChanged ()
```

8.12.2.2 getAddress()

```
QString DeviceInfo::getAddress () const
```

8.12.2.3 getDevice()

```
QBluetoothDeviceInfo DeviceInfo::getDevice ()
```

8.12.2.4 getName()

```
QString DeviceInfo::getName () const
```

8.12.2.5 setDevice()

```
void DeviceInfo::setDevice (
    const QBluetoothDeviceInfo & dev)
```

8.12.3 Property Documentation

8.12.3.1 deviceName

```
QString DeviceInfo::deviceName [read]
```

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

- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header/generic/[deviceinfo.h](#)
- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/generic/[deviceinfo.cpp](#)

8.13 GeoidHelper Class Reference

```
#include <geoid_helper.h>
```

Data Structures

- struct [Impl](#)

Public Member Functions

- [GeoidHelper](#) (const std::string &geoidDataPath="")
- std::optional< [GeoidResult](#) > [compensatedHeight](#) (double lat_deg, double lon_deg, double h_ellipsoid_m, double alpha=1.0)
- bool [isValid](#) () const

8.13.1 Constructor & Destructor Documentation

8.13.1.1 GeoidHelper()

```
GeoidHelper::GeoidHelper (
    const std::string & geoidDataPath = "")
```

8.13.2 Member Function Documentation

8.13.2.1 compensatedHeight()

```
std::optional< GeoidResult > GeoidHelper::compensatedHeight (
    double lat_deg,
    double lon_deg,
    double h_ellipsoid_m,
    double alpha = 1.0)
```

8.13.2.2 isValid()

```
bool GeoidHelper::isValid () const
```

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

- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header/driver/geoid_helper.h
- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/driver/geoid_helper.cpp

8.14 GeoidResult Struct Reference

```
#include <geoid_helper.h>
```

Data Fields

- double N
- double h_compensated

8.14.1 Field Documentation

8.14.1.1 h_compensated

```
double GeoidResult::h_compensated
```

8.14.1.2 N

```
double GeoidResult::N
```

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

- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header/driver/[geoid_helper.h](#)

8.15 GpxParser Class Reference

```
#include <gpx_parse.h>
```

Public Member Functions

- bool [parseFile](#) (const QString &filePath)
- QList< [TrackPoint](#) > [getTrackPoints](#) () const

8.15.1 Member Function Documentation

8.15.1.1 [getTrackPoints\(\)](#)

```
QList< TrackPoint > GpxParser::getTrackPoints () const
```

8.15.1.2 [parseFile\(\)](#)

```
bool GpxParser::parseFile (
    const QString & filePath)
```

Here is the call graph for this function:



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

- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header/driver/[gpx_parse.h](#)
- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/driver/[gpx_parse.cpp](#)

8.16 GeoidHelper::Impl Struct Reference

Data Fields

- bool [valid](#) = false

8.16.1 Field Documentation

8.16.1.1 valid

```
bool GeoidHelper::Impl::valid = false
```

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

- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/driver/[geoid_helper.cpp](#)

8.17 KeepAwakeHelper Class Reference

```
#include <lockhelper.h>
```

Public Member Functions

- [KeepAwakeHelper \(\)](#)
- void [EnableKeepAwakeHelper \(\)](#)
- virtual [~KeepAwakeHelper \(\)](#)

8.17.1 Constructor & Destructor Documentation

8.17.1.1 KeepAwakeHelper()

```
KeepAwakeHelper::KeepAwakeHelper ()
```

8.17.1.2 ~KeepAwakeHelper()

```
KeepAwakeHelper::~KeepAwakeHelper () [virtual]
```

8.17.2 Member Function Documentation

8.17.2.1 EnableKeepAwakeHelper()

```
void KeepAwakeHelper::EnableKeepAwakeHelper ()
```

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

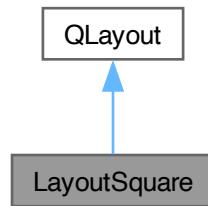
- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header/generic/[lockhelper.h](#)
- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/generic/[lockhelper.cpp](#)

8.18 LayoutSquare Class Reference

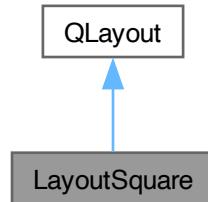
This class provides constant aspect ration (square) layout for a single widget.

```
#include <LayoutSquare.h>
```

Inheritance diagram for LayoutSquare:



Collaboration diagram for LayoutSquare:



Public Member Functions

- [LayoutSquare](#) (QWidget *parent, int spacing=-1)
Constructor.
- [LayoutSquare](#) (int spacing=-1)
Constructor.
- virtual [~LayoutSquare](#) ()
Destructor.
- void [addItem](#) (QLayoutItem *item)
- void [addWidget](#) (QWidget *widget)
- int [count](#) () const
- Qt::Orientations [expandingDirections](#) () const
- QRect [geometry2](#) ()
- bool [hasHeightForWidth](#) () const

- bool `hasItem () const`
- QLayoutItem * `itemAt` (int index) const
- QSize `minimumSize () const`
- QLayoutItem * `replaceItem` (QLayoutItem *item)
- void `setGeometry` (const QRect &rect)
- QSize `sizeHint () const`
- QLayoutItem * `take ()`
- QLayoutItem * `takeAt` (int index)

8.18.1 Detailed Description

This class provides constant aspect ration (square) layout for a single widget.

8.18.2 Constructor & Destructor Documentation

8.18.2.1 LayoutSquare() [1/2]

```
LayoutSquare::LayoutSquare (
    QWidget * parent,
    int spacing = -1) [explicit]
```

Constructor.

8.18.2.2 LayoutSquare() [2/2]

```
LayoutSquare::LayoutSquare (
    int spacing = -1) [explicit]
```

Constructor.

8.18.2.3 ~LayoutSquare()

```
LayoutSquare::~LayoutSquare () [virtual]
```

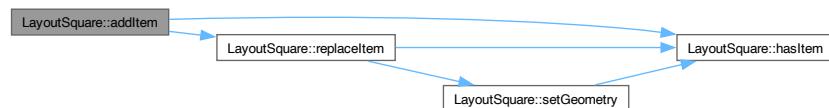
Destructor.

8.18.3 Member Function Documentation

8.18.3.1 addItem()

```
void LayoutSquare::addItem (
    QLayoutItem * item)
```

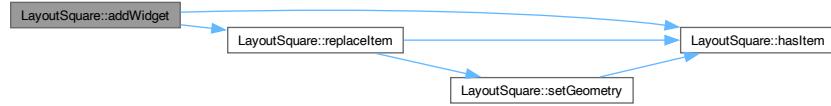
Here is the call graph for this function:



8.18.3.2 addWidget()

```
void LayoutSquare::addWidget (QWidget * widget)
```

Here is the call graph for this function:



8.18.3.3 count()

```
int LayoutSquare::count () const
```

Here is the call graph for this function:



8.18.3.4 expandingDirections()

```
Qt::Orientations LayoutSquare::expandingDirections () const
```

8.18.3.5 geometry2()

```
QRect LayoutSquare::geometry2 ()
```

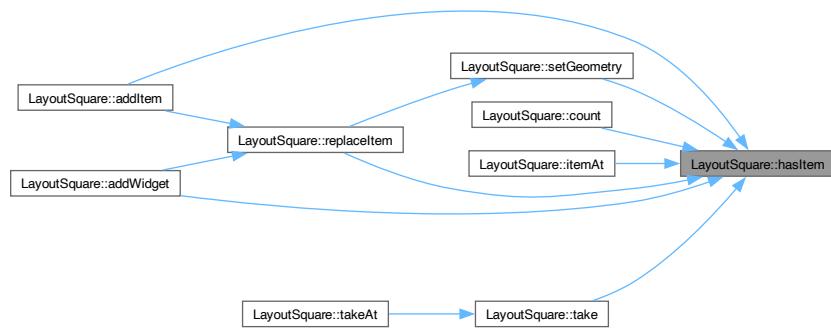
8.18.3.6 hasHeightForWidth()

```
bool LayoutSquare::hasHeightForWidth () const
```

8.18.3.7 hasItem()

```
bool LayoutSquare::hasItem () const
```

Here is the caller graph for this function:



8.18.3.8 itemAt()

```
QLayoutItem * LayoutSquare::itemAt (
    int index) const
```

Here is the call graph for this function:



8.18.3.9 minimumSize()

```
QSize LayoutSquare::minimumSize () const
```

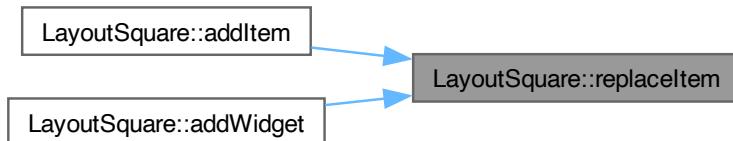
8.18.3.10 replaceItem()

```
QLayoutItem * LayoutSquare::replaceItem (
    QLayoutItem * item)
```

Here is the call graph for this function:



Here is the caller graph for this function:



8.18.3.11 setGeometry()

```
void LayoutSquare::setGeometry (const QRect & rect)
```

Here is the call graph for this function:



Here is the caller graph for this function:



8.18.3.12 sizeHint()

```
QSize LayoutSquare::sizeHint () const
```

8.18.3.13 take()

```
QLayoutItem * LayoutSquare::take ()
```

Here is the call graph for this function:

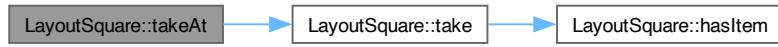


Here is the caller graph for this function:

**8.18.3.14 takeAt()**

```
QLayoutItem * LayoutSquare::takeAt (
    int index)
```

Here is the call graph for this function:



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

- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/LayoutSquare.h
- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/LayoutSquare.cpp

8.19 LFD Struct Reference

Data Fields

- double L
- double F
- double D

8.19.1 Field Documentation

8.19.1.1 D

```
double LFD::D
```

8.19.1.2 F

```
double LFD::F
```

8.19.1.3 L

```
double LFD::L
```

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

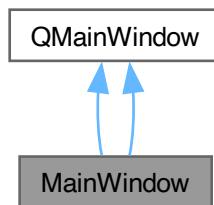
- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/generic/[ins_driver.cpp](#)
- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/generic/[mytcpsocket.cpp](#)

8.20 MainWindow Class Reference

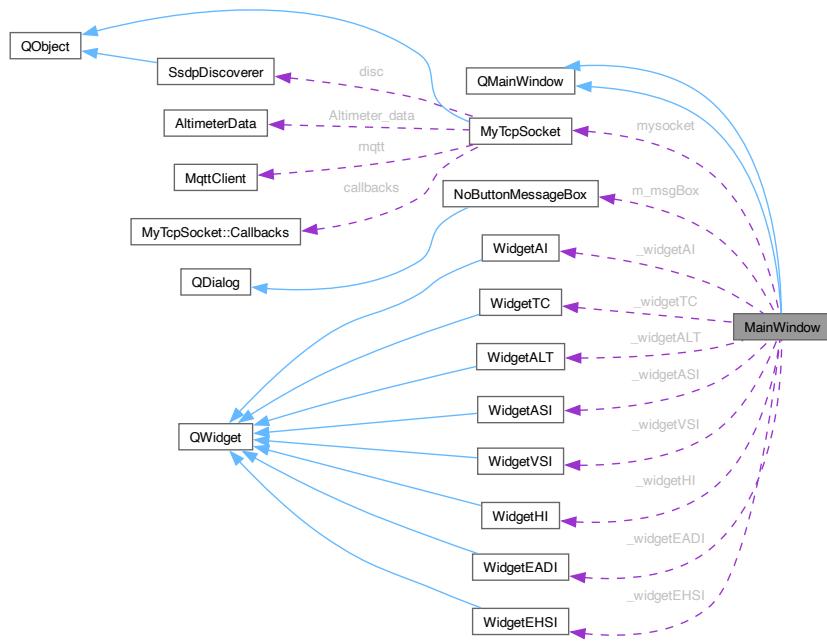
Main UI window for the Glasscockpit 200-UAV.

```
#include <mainwindow.h>
```

Inheritance diagram for MainWindow:



Collaboration diagram for MainWindow:



Public Member Functions

- [MainWindow \(QWidget *parent=nullptr\)](#)
- [~MainWindow \(\)](#)
- void [addnext \(int x\)](#)
Shift next transponder code and append a digit (0..7).
- void [addcurrent \(int x\)](#)
(Unused helper, kept for compatibility) Add to current code.
- void [setmode \(int mode\)](#)
Set transponder mode and send to hardware via [MyTcpSocket](#).
- int [set_default_radio \(void\)](#)
Initialize radio list with default values and save to file.
- int [set_default_planes \(void\)](#)
Initialize airplane list with default values and save to file.
- int [set_default_config \(const Matrix3x6 &sensor\)](#)
Serialize calibration config (Accel/Gyro/Mag + IDs) to CONFIG file.
- int [get_default_config \(Matrix3x6 &sensor\)](#)
Load calibration config from CONFIG file.
- void [logLanded \(\)](#)
Log landing time + duration to log and UI.
- void [logTakeoff \(\)](#)
Log takeoff time to log and UI.
- void [init \(\)](#)
Initialize camera and permissions (called from ctor).
- void [updateCameras \(\)](#)
Refresh available camera list.

- void `hideCamera ()`
Stop and hide camera preview.
- void `setCamera (const QCameraDevice &cameraDevice)`
Select active camera and wire it into viewfinder / capture.
- void `permissionUpdated (const QPermission &permission)`
Called when location permission request resolves.
- void `setButtonIcon (QString iconPath, QPushButton *button)`
Helper to set a pixmap icon on a QPushButton.
- double `getBearing (double lat1, double lon1, double lat2, double lon2)`
Compute initial course (bearing) from (lat1,lon1) to (lat2,lon2).
- void `AccelerometerRead ()`
Calibration helper: update gravity estimate from accelerometer.
- double `setQNH ()`
Compute and apply QNH to match baro altitude with GPS altitude.
- void `showImage ()`
Load startup logo image and configure "fly home" button.
- void `onResized (int)`
(Unused) react to window resize.
- QScreen * `getActiveScreen (QWidget *pWidget) const`
Return QScreen on which the widget is currently shown.
- `WidgetAI * getAI ()`
- `WidgetTC * getTC ()`
- `WidgetALT * getALT ()`
- `WidgetASI * getASI ()`
- `WidgetVSI * getVSI ()`
- `WidgetHI * getHI ()`
- `WidgetEADI * getEADI ()`
- `WidgetEHSI * getEHSI ()`
- `MainWindow (QWidget *parent=Q_NULLPTR)`
- `~MainWindow ()`

Static Public Member Functions

- static void `setIMU (void *parent, bool use_imu)`
C callback from MyTcpSocket when IMU presence is known.

Data Fields

- QSplashScreen * `splash` = nullptr
Optional iOS splash screen shown at startup.
- Vector3d `m_attitude`
Current attitude estimate [roll, pitch, yaw] in degrees.
- double `roll_blended` = 0.0
Blended roll estimate (IMU + GPS turn-rate).
- bool `roll_blended_ok` = false
- Matrix3x3 `rotationMatrix`
Optional rotation matrix (not heavily used yet).
- Qt::ScreenOrientation `ScreenMode`
Screen orientation (portrait/landscape).
- int `next [4]` = {7, 0, 0, 0}

- int **current** [4] = {8, 8, 8, 8}
 - Transponder code being set (digits 0..7).*
- int **mode** = 0
 - Transponder active code.*
- int **m_install**
 - IMU mounting orientation (tilt) used for compensation.*
- double **m_vario** = 0.0
 - Instantaneous altitude change (delta) for vario calculation.*
- double **takeoff_latitude** = 0.0
 - GPS position at takeoff.*
- double **takeoff_longitude** = 0.0
- double **takeoff_altitude** = 0.0
- double **m_tansALT** = 0.0
 - Transponder-reported altitude (feet or meters depending on mode).*
- bool **m_use_imu** = false
 - True if an external IMU (via [MyTcpSocket](#)) is used instead of phone sensors.*
- double **m_head** = 9999.0
 - Ground speed [km/h].*
- qreal **m_temp** = 9999.0
 - Ambient temperature [°C].*
- double **m_roll_angle** = 0.0
 - Roll angle derived from GPS turn rate (rad).*
- double **m_total_accel** = 0.0
 - Net acceleration magnitude.*
- double **m_var_speed** = 0.0
 - Vertical speed [ft/min] filtered.*
- double **m_ms** = 0.0
 - Misc time variable (seconds).*
- int **Radar_depth** = 120
 - Radar depth (number of scan "columns").*
- int **Radar_Height** = 120
 - Radar height scaling.*
- bool **m_bluetoothrunning** = false
 - Used during startup while BT scanning / IMU connection is in progress.*
- QDateTime **m_takeoffTime**
 - Takeoff and landing timestamps.*
- QDateTime **m_landedTime**
- MyTcpSocket * **mysocket** = nullptr
 - Main hardware IO handler (transponder / radar / INS / MQTT).*
- int **currentIndex** = 0
 - Current index of stacked widget.*
- QString **_transponder_id** = "4150323833373009"
 - Default transponder ID (SIM).*
- QString **_radar_id** = "415032383337320B"
 - Default radar ID (SIM).*
- QString **_IMU_id** = "4150323833373205"
- NoButtonMessageBox * **m_msgBox** = nullptr
 - Transient "please wait" message during startup.*
- QQuickView **view**
- QSplineSeries * **series** = nullptr

- QElapsedTimer `m_timer`
- WidgetAI * `_widgetAI` = nullptr
- WidgetTC * `_widgetTC` = nullptr
- WidgetALT * `_widgetALT` = nullptr
- WidgetASI * `_widgetASI` = nullptr
- WidgetVSI * `_widgetVSI` = nullptr
- WidgetHI * `_widgetHI` = nullptr
- WidgetEADI * `_widgetEADI` = nullptr
- WidgetEHSI * `_widgetEHSI` = nullptr
- int `screen_index` = 0
Which screen (monitor) index we are using.
- Ui::SCREEN * `ui` = (Ui::SCREEN *) &(*new (Ui::SCREEN))

Protected Member Functions

- void `timerEvent` (QTimerEvent *event)

8.20.1 Detailed Description

Main UI window for the Glasscockpit 200-UAV.

The [MainWindow](#) class.

Responsibilities:

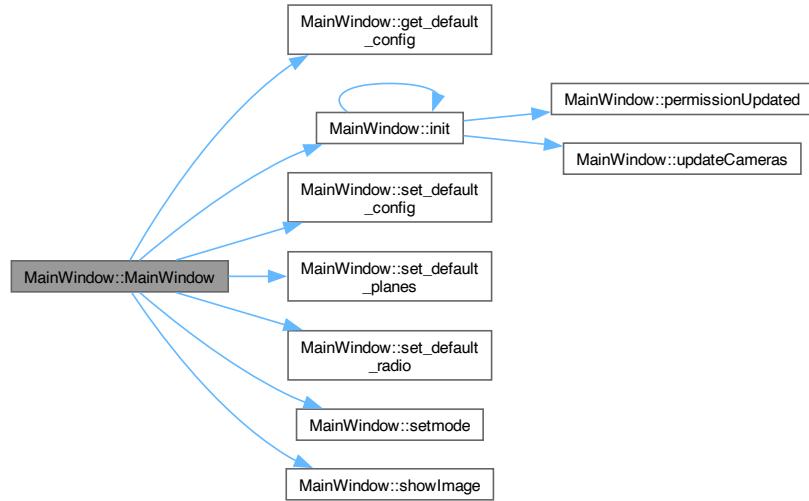
- Manage pages (transponder, IMU, primary instruments, glass cockpit, radar, radio list, autopilot, config, camera).
- Own and configure [MyTcpSocket](#) (transponder, IMU/INS, radar, MQTT).
- Interface with onboard sensors (accelerometer, gyro, magnetometer, baro, orientation, temperature) when external IMU is not present.
- Run EKF-based attitude/heading estimation and propagate to widgets.
- Handle camera capture, GPS updates, and simple flight logging.

8.20.2 Constructor & Destructor Documentation

8.20.2.1 MainWindow() [1/2]

```
MainWindow::MainWindow (
    QWidget * parent = nullptr) [explicit]
```

Here is the call graph for this function:



Here is the caller graph for this function:



8.20.2.2 ~MainWindow() [1/2]

```
MainWindow::~MainWindow ()
```

8.20.2.3 MainWindow() [2/2]

```
MainWindow::MainWindow (
    QWidget * parent = Q_NULLPTR) [explicit]
```

Constructor.

8.20.2.4 ~MainWindow() [2/2]

```
MainWindow::~MainWindow ()
```

Destructor.

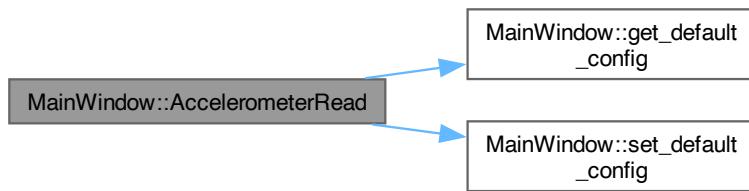
8.20.3 Member Function Documentation

8.20.3.1 AccelerometerRead()

```
void MainWindow::AccelerometerRead ()
```

Calibration helper: update gravity estimate from accelerometer.

Here is the call graph for this function:



8.20.3.2 addcurrent()

```
void MainWindow::addcurrent (
    int x)
```

(Unused helper, kept for compatibility) Add to current code.

8.20.3.3 addnext()

```
void MainWindow::addnext (
    int x)
```

Shift next transponder code and append a digit (0..7).

8.20.3.4 get_default_config()

```
int MainWindow::get_default_config (
    Matrix3x6 & sensor)
```

Load calibration config from CONFIG file.

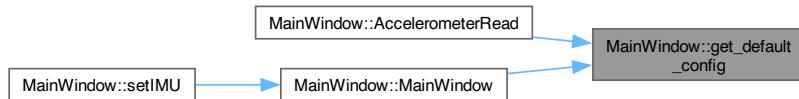
Parameters

<code>sensor</code>	[out] 3x6 calibration matrix.
---------------------	-------------------------------

Returns

0 on success, negative on error (e.g. missing entries).

Here is the caller graph for this function:

**8.20.3.5 `getActiveScreen()`**

```
QScreen * MainWindow::getActiveScreen (
    QWidget * pWidget) const
```

Return QScreen on which the widget is currently shown.

8.20.3.6 `getAI()`

```
WidgetAI * MainWindow::getAI () [inline]
```

8.20.3.7 `getALT()`

```
WidgetALT * MainWindow::getALT () [inline]
```

8.20.3.8 `getASI()`

```
WidgetASI * MainWindow::getASI () [inline]
```

8.20.3.9 `getBearing()`

```
double MainWindow::getBearing (
    double lat1,
    double lon1,
    double lat2,
    double lon2)
```

Compute initial course (bearing) from (lat1,lon1) to (lat2,lon2).

Returns

bearing in degrees (0 = North, 90 = East).

8.20.3.10 getEADI()

```
WidgetEADI * MainWindow::getEADI () [inline]
```

8.20.3.11 getEHJSI()

```
WidgetEHJSI * MainWindow::getEHJSI () [inline]
```

8.20.3.12 getHI()

```
WidgetHI * MainWindow::getHI () [inline]
```

8.20.3.13 getTC()

```
WidgetTC * MainWindow::getTC () [inline]
```

8.20.3.14 getVSI()

```
WidgetVSI * MainWindow::getVSI () [inline]
```

8.20.3.15 hideCamera()

```
void MainWindow::hideCamera ()
```

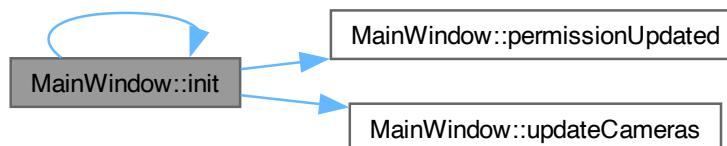
Stop and hide camera preview.

8.20.3.16 init()

```
void MainWindow::init ()
```

Initialize camera and permissions (called from ctor).

Here is the call graph for this function:



Here is the caller graph for this function:



8.20.3.17 logLanded()

```
void MainWindow::logLanded ()
```

Log landing time + duration to log and UI.

8.20.3.18 logTakeoff()

```
void MainWindow::logTakeoff ()
```

Log takeoff time to log and UI.

8.20.3.19 onResized()

```
void MainWindow::onResized (
    int )
```

(Unused) react to window resize.

8.20.3.20 permissionUpdated()

```
void MainWindow::permissionUpdated (
    const QPermission & permission)
```

Called when location permission request resolves.

Here is the caller graph for this function:

**8.20.3.21 set_default_config()**

```
int MainWindow::set_default_config (
    const Matrix3x6 & sensor)
```

Serialize calibration config (Accel/Gyro/Mag + IDs) to CONFIG file.

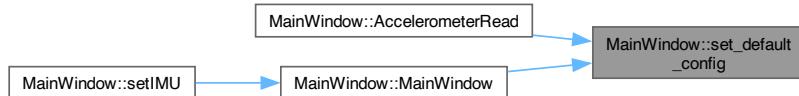
Parameters

<i>sensor</i>	3x6 calibration matrix (Accel, Gyro, Mag rows).
---------------	---

Returns

0 on success, -1 on error.

Here is the caller graph for this function:

**8.20.3.22 set_default_planes()**

```
int MainWindow::set_default_planes (
    void )
```

Initialize airplane list with default values and save to file.

Returns

0 on success, -1 on error.

Here is the caller graph for this function:

**8.20.3.23 set_default_radio()**

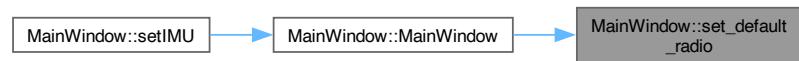
```
int MainWindow::set_default_radio (
    void )
```

Initialize radio list with default values and save to file.

Returns

0 on success, -1 on error.

Here is the caller graph for this function:



8.20.3.24 setButtonIcon()

```
void MainWindow::setButtonIcon (
    QString iconPath,
    QPushButton * button)
```

Helper to set a pixmap icon on a QPushButton.

8.20.3.25 setCamera()

```
void MainWindow::setCamera (
    const QCameraDevice & cameraDevice)
```

Select active camera and wire it into viewfinder / capture.

8.20.3.26 setIMU()

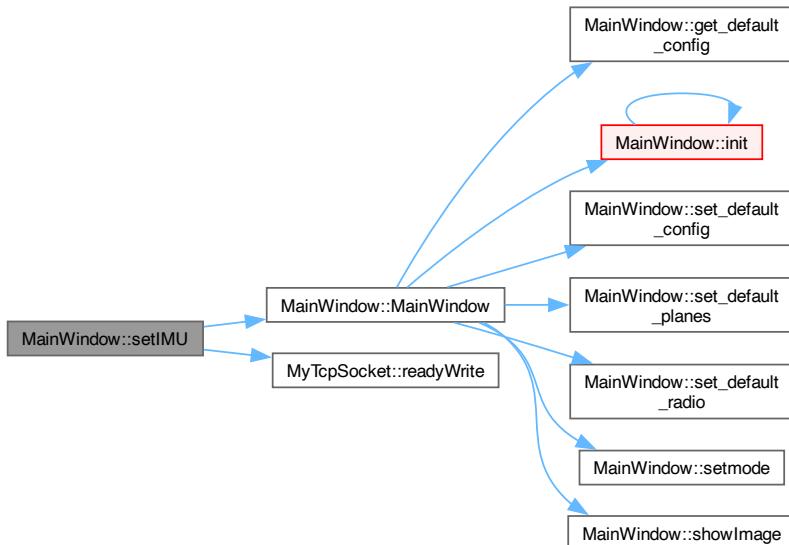
```
void MainWindow::setIMU (
    void * parent,
    bool use_imu) [static]
```

C callback from [MyTcpSocket](#) when IMU presence is known.

Parameters

<i>parent</i>	Pointer back to MainWindow instance.
<i>use_imu</i>	True if an external IMU is available.

Here is the call graph for this function:

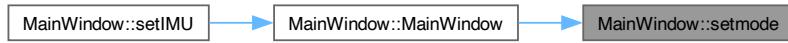


8.20.3.27 setmode()

```
void MainWindow::setmode (
    int mode)
```

Set transponder mode and send to hardware via [MyTcpSocket](#).

Mode: 0 = query (s=?) 1 = STBY 2 = ALT ON 3 = ALT/ident Here is the caller graph for this function:



8.20.3.28 setQNH()

```
double MainWindow::setQNH ()
```

Compute and apply QNH to match baro altitude with GPS altitude.

Returns

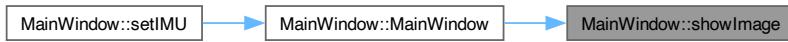
QNH in hPa.

8.20.3.29 showImage()

```
void MainWindow::showImage ()
```

Load startup logo image and configure "fly home" button.

Here is the caller graph for this function:



8.20.3.30 timerEvent()

```
void MainWindow::timerEvent (
    QTimerEvent * event) [protected]
```

Timer timeout callback. This function is called on default timer timeout event.

8.20.3.31 updateCameras()

```
void MainWindow::updateCameras ()
```

Refresh available camera list.

Here is the caller graph for this function:



8.20.4 Field Documentation

8.20.4.1 _IMU_id

```
QString MainWindow::_IMU_id = "4150323833373205"
```

8.20.4.2 _radar_id

```
QString MainWindow::_radar_id = "415032383337320B"
```

Default radar ID (SIM).

8.20.4.3 _transponder_id

```
QString MainWindow::_transponder_id = "4150323833373009"
```

Default transponder ID (SIM).

8.20.4.4 _widgetAI

```
WidgetAI* MainWindow::_widgetAI = nullptr
```

8.20.4.5 _widgetALT

```
WidgetALT* MainWindow::_widgetALT = nullptr
```

8.20.4.6 _widgetASI

```
WidgetASI* MainWindow::_widgetASI = nullptr
```

8.20.4.7 `_widgetEADI`

```
WidgetEADI* MainWindow::_widgetEADI = nullptr
```

8.20.4.8 `_widgetEHSI`

```
WidgetEHSI* MainWindow::_widgetEHSI = nullptr
```

8.20.4.9 `_widgetHI`

```
WidgetHI* MainWindow::_widgetHI = nullptr
```

8.20.4.10 `_widgetTC`

```
WidgetTC* MainWindow::_widgetTC = nullptr
```

8.20.4.11 `_widgetVSI`

```
WidgetVSI* MainWindow::_widgetVSI = nullptr
```

8.20.4.12 `current`

```
int MainWindow::current[4] = {8, 8, 8, 8}
```

Transponder active code.

8.20.4.13 `currentIndex`

```
int MainWindow::currentIndex = 0
```

Current index of stacked widget.

8.20.4.14 `m_attitude`

```
Vector3d MainWindow::m_attitude
```

Current attitude estimate [roll, pitch, yaw] in degrees.

8.20.4.15 `m_bluetoothrunning`

```
bool MainWindow::m_bluetoothrunning = false
```

Used during startup while BT scanning / IMU connection is in progress.

8.20.4.16 m_head

```
double MainWindow::m_head = 9999.0
```

Ground speed [km/h].

Heading used by instruments (deg).

8.20.4.17 m_install

```
Vector3d MainWindow::m_install
```

IMU mounting orientation (tilt) used for compensation.

8.20.4.18 m_landedTime

```
QDateTime MainWindow::m_landedTime
```

8.20.4.19 m_ms

```
double MainWindow::m_ms = 0.0
```

Misc time variable (seconds).

8.20.4.20 m_msgBox

```
NoButtonMessageBox* MainWindow::m_msgBox = nullptr
```

Transient "please wait" message during startup.

8.20.4.21 m_roll_angle

```
double MainWindow::m_roll_angle = 0.0
```

Roll angle derived from GPS turn rate (rad).

8.20.4.22 m_takeoffTime

```
QDateTime MainWindow::m_takeoffTime
```

Takeoff and landing timestamps.

8.20.4.23 m_tansALT

```
double MainWindow::m_tansALT = 0.0
```

Transponder-reported altitude (feet or meters depending on mode).

8.20.4.24 m_temp

```
qreal MainWindow::m_temp = 9999.0
```

Ambient temperature [°C].

8.20.4.25 m_timer

```
QEapsedTimer MainWindow::m_timer
```

8.20.4.26 m_total_accel

```
double MainWindow::m_total_accel = 0.0
```

Net acceleration magnitude.

8.20.4.27 m_use_imu

```
bool MainWindow::m_use_imu = false
```

True if an external IMU (via [MyTcpSocket](#)) is used instead of phone sensors.

8.20.4.28 m_var_speed

```
double MainWindow::m_var_speed = 0.0
```

Vertical speed [ft/min] filtered.

8.20.4.29 m_vario

```
double MainWindow::m_vario = 0.0
```

Instantaneous altitude change (delta) for vario calculation.

8.20.4.30 mode

```
int MainWindow::mode = 0
```

Current transponder mode (0..3).

8.20.4.31 mysocket

```
MyTcpSocket* MainWindow::mysocket = nullptr
```

Main hardware IO handler (transponder / radar / INS / MQTT).

8.20.4.32 next

```
int MainWindow::next[4] = {7, 0, 0, 0}
```

Transponder code being set (digits 0..7).

8.20.4.33 Radar_depth

```
int MainWindow::Radar_depth = 120
```

Radar depth (number of scan "columns").

8.20.4.34 Radar_Height

```
int MainWindow::Radar_Height = 120
```

Radar height scaling.

8.20.4.35 roll_blended

```
double MainWindow::roll_blended = 0.0
```

Blended roll estimate (IMU + GPS turn-rate).

8.20.4.36 roll_blended_ok

```
bool MainWindow::roll_blended_ok = false
```

8.20.4.37 rotationMatrix

[Matrix3x3](#) MainWindow::rotationMatrix

Optional rotation matrix (not heavily used yet).

8.20.4.38 screen_index

```
int MainWindow::screen_index = 0
```

Which screen (monitor) index we are using.

8.20.4.39 ScreenMode

```
Qt::ScreenOrientation MainWindow::ScreenMode
```

Screen orientation (portrait/landscape).

8.20.4.40 series

```
QSplineSeries* MainWindow::series = nullptr
```

8.20.4.41 splash

```
QSplashScreen* MainWindow::splash = nullptr
```

Optional iOS splash screen shown at startup.

8.20.4.42 takeoff_altitude

```
double MainWindow::takeoff_altitude = 0.0
```

8.20.4.43 takeoff_latitude

```
double MainWindow::takeoff_latitude = 0.0
```

GPS position at takeoff.

8.20.4.44 takeoff_longitude

```
double MainWindow::takeoff_longitude = 0.0
```

8.20.4.45 ui

```
Ui::SCREEN* MainWindow::ui = (Ui::SCREEN *) &(*new (Ui::SCREEN))
```

Qt Designer-generated UI class instance. NOTE: constructed in a custom way; left unchanged for compatibility.

8.20.4.46 view

```
QQuickView MainWindow::view
```

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

- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header/generic/mainwindow.h
- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/MainWindow.h
- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/generic/mainwindow.cpp
- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/MainWindow.cpp

8.21 MqttClient Class Reference

Wrapper around mqtt::async_client for "v3-style" MQTT usage.

```
#include <mqttclient.h>
```

Public Types

- using `MsgHandler = std::function<void(const std::string&, const std::string&)>`
User message handler type: handler(topic, payload).

Public Member Functions

- `MqttClient (const std::string &server, const std::string &clientId)`
Construct MQTT client wrapper.
- `void connect (bool cleanSession=true, std::chrono::seconds keepAlive=std::chrono::seconds{20})`
Connect to the broker (v3 style).
- `void disconnect ()`
Disconnect from the broker.
- `void sendMessage (const std::string &topic, const std::string &payload, int qos=1, bool retained=false)`
Publish a message.
- `void subscribe (const std::string &topic, int qos=1)`
Subscribe to a topic.
- `void setMessageHandler (MsgHandler handler)`
Register user message handler.
- `mqtt::async_client & client ()`
Direct access to underlying async_client (advanced use only).

8.21.1 Detailed Description

Wrapper around mqtt::async_client for "v3-style" MQTT usage.

This version avoids v5-only types (no properties or reason codes) and exposes:

- `connect(cleanSession, keepAlive)`
- `disconnect()`
- `sendMessage(topic, payload, qos, retained)`
- `subscribe(topic, qos)`

Incoming messages are forwarded to a user-provided `MsgHandler`.

8.21.2 Member Typedef Documentation

8.21.2.1 MsgHandler

```
using MqttClient::MsgHandler = std::function<void(const std::string&, const std::string&)>
```

User message handler type: `handler(topic, payload)`.

8.21.3 Constructor & Destructor Documentation

8.21.3.1 MqttClient()

```
MqttClient::MqttClient (
    const std::string & server,
    const std::string & clientId)
```

Construct MQTT client wrapper.

Parameters

<i>server</i>	Broker URI, e.g. "tcp://localhost:1883".
<i>clientId</i>	Client identifier to use when connecting.

8.21.4 Member Function Documentation

8.21.4.1 client()

```
mqtt::async_client & MqttClient::client () [inline]
```

Direct access to underlying `async_client` (advanced use only).

8.21.4.2 connect()

```
void MqttClient::connect (
    bool cleanSession = true,
    std::chrono::seconds keepAlive = std::chrono::seconds{20})
```

Connect to the broker (v3 style).

Parameters

<i>cleanSession</i>	Whether to request a clean session.
<i>keepAlive</i>	Keep-alive interval.

8.21.4.3 disconnect()

```
void MqttClient::disconnect ()
```

Disconnect from the broker.

8.21.4.4 sendMessage()

```
void MqttClient::sendMessage (
    const std::string & topic,
    const std::string & payload,
    int qos = 1,
    bool retained = false)
```

Publish a message.

Parameters

<i>topic</i>	Topic name.
<i>payload</i>	Message payload.
<i>qos</i>	QoS level (0, 1, or 2).
<i>retained</i>	Retained flag.

8.21.4.5 setMessageHandler()

```
void MqttClient::setMessageHandler (
    MsgHandler handler) [inline]
```

Register user message handler.

Called whenever a new message arrives on subscribed topics.

8.21.4.6 subscribe()

```
void MqttClient::subscribe (
    const std::string & topic,
    int qos = 1)
```

Subscribe to a topic.

Parameters

<i>topic</i>	Topic filter to subscribe.
<i>qos</i>	Desired QoS for subscription.

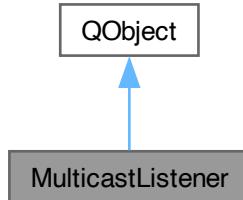
The documentation for this class was generated from the following files:

- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header/driver/[mqttclient.h](#)
- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/driver/[mqttclient.cpp](#)

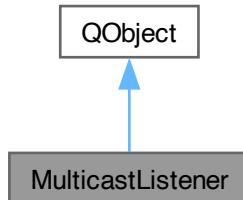
8.22 MulticastListener Class Reference

```
#include <multicastlistner.h>
```

Inheritance diagram for MulticastListener:



Collaboration diagram for MulticastListener:



Public Member Functions

- `MulticastListener (QObject *parent=nullptr)`
- `~MulticastListener () override`
- `bool start (const QString &groupAddr=QStringLiteral("239.255.0.1"), quint16 port=4210)`
- `void stop ()`
- `void messageReceived (const QString &message, const QHostAddress &sender, quint16 senderPort)`
- `void errorOccurred (const QString &error)`

8.22.1 Constructor & Destructor Documentation

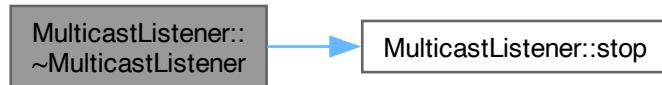
8.22.1.1 MulticastListener()

```
MulticastListener::MulticastListener (
    QObject * parent = nullptr) [explicit]
```

8.22.1.2 ~MulticastListener()

```
MulticastListener::~MulticastListener () [override]
```

Here is the call graph for this function:

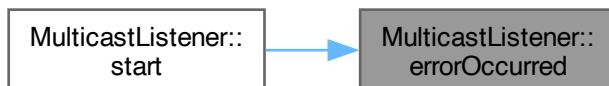


8.22.2 Member Function Documentation

8.22.2.1 errorOccurred()

```
void MulticastListener::errorOccurred (\n    const QString & error)
```

Here is the caller graph for this function:



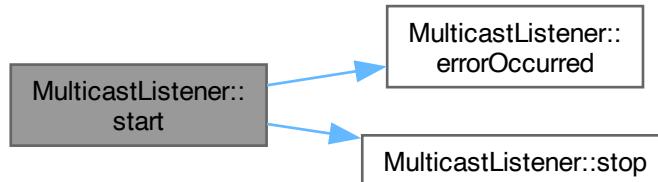
8.22.2.2 messageReceived()

```
void MulticastListener::messageReceived (\n    const QString & message,\n    const QHostAddress & sender,\n    quint16 senderPort)
```

8.22.2.3 start()

```
bool MulticastListener::start (
    const QString & groupAddr = QStringLiteral("239.255.0.1"),
    quint16 port = 4210)
```

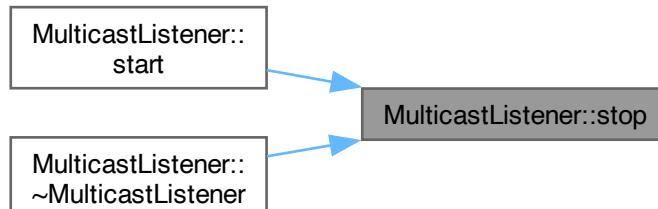
Here is the call graph for this function:



8.22.2.4 stop()

```
void MulticastListener::stop ()
```

Here is the caller graph for this function:



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

- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header/generic/multicastlistner.h
- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/generic/multicastlistner.cpp

8.23 MyDevice Class Reference

```
#include <MyDevice.h>
```

Public Member Functions

- [MyDevice \(\)](#)

8.23.1 Constructor & Destructor Documentation

8.23.1.1 MyDevice()

```
MyDevice::MyDevice ()
```

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

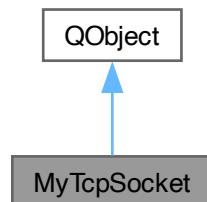
- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header/generic/[MyDevice.h](#)
- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/generic/[MyDevice.cpp](#)

8.24 MyTcpSocket Class Reference

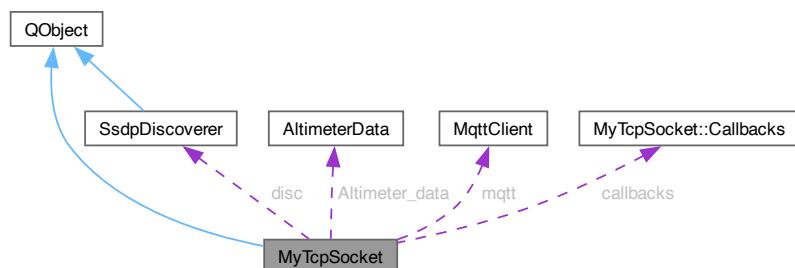
Handles communication with:

```
#include <mytcpsocket.h>
```

Inheritance diagram for MyTcpSocket:



Collaboration diagram for MyTcpSocket:



Data Structures

- struct [Callbacks](#)

Public Types

- `typedef struct MyTcpSocket::Callbacks Callbacks`

Public Member Functions

- `MyTcpSocket` (QObject *parent=nullptr, QPlainTextEdit *s=nullptr, void(*retx)(void *, const char *, uint32_t)=nullptr, void(*retry)(void *, bool use_imu)=nullptr)
Construct a [MyTcpSocket](#).
- `~MyTcpSocket ()`
Destructor. Closes all serial ports.
- `void ssdpConfig ()`
multicastConfig. Sends multicast...
- `void readyWrite (char *data)`
Send a raw ASCII command to the transponder, if connected.
- `void doConnect ()`
Initiate TCP/UDP connection if used (currently unused/stub).
- `void connected ()`
Try to connect and initialize the transponder.
- `void connectedIMU ()`
Try to connect and initialize IMU (BT and/or USB) and Radar.
- `void connectedRadar ()`
Try to connect and initialize RADAR (Net or USB).
- `void connectedAltitude ()`
Try to connect and initialize RADAR (Net or USB).
- `void setbacklit ()`
Android: periodically bump external display backlight to max.
- `void setSerialPorts (QString imu, QString transponder, QString radar)`
Configure which devices to look for by "serial ID" or logical ID.
- `void handleUpdate (const std::string &ID, const std::string &value)`
Process an incoming MQTT message from the X-Plane bridge.
- `void parseAltimeterLine (const QString &line)`
- `int com_setup (QSerialPort *com_port, QString sport)`
Configure a QSerialPort with given port name.
- `void CallSwiftMemberFromC (void *classPtr, void(*callback)(void *))`
Example bridge to invoke Swift member functions from C code.
- `void sendMessage (const QString &message)`
Generic Qt signal for sending status / log messages to UI.
- `void doTransponder ()`
Periodic transponder polling / configuration state machine.
- `void doStart ()`
Startup state machine handler.

Static Public Member Functions

- static void `parseIMU` (void *parent, uint32_t uiReg, uint16_t sReg[])
Static callback for IMU data (called from C driver).
- static void `doRadar` (void *parent, const char *data, uint32_t length)
Static callback for Radar data (called from C driver).

Data Fields

- void(* `ret_imu`) (void *, bool use_imu) = nullptr
Callback to report IMU usage/availability to external C code.
- void(* `ret_transponder`) (void *, const char *data, uint32_t size) = nullptr
Callback invoked when serial data arrives from transponder.
- `AltimeterData Altimeter_data` = {0,0,0,0}
- `QString m_address` = "239.255.0.1"
- `quint16 MCAST_PORT` = 4210
- `QSerialPort * port` = nullptr
Generic serial port (unused here; legacy).
- `QSerialPort * lidar` = nullptr
Optional lidar serial port (if used elsewhere).
- `QList< QSerialPortInfo > serialport`
Cached list of available serial ports.
- `SsdpDiscoverer * disc` = nullptr
- `QString sport`
Current serial port name (used by some legacy code).
- `QPlainTextEdit * text` = nullptr
Optional text log widget.
- `QString _transponder_id`
Device identifiers (serial numbers / logical IDs).
- `QString _radar_id`
- `QString _IMU_id`
- `std::string SERVER_ADDRESS`
MQTT broker address (e.g. "tcp://localhost:1883").
- `std::string CLIENT_ID`
MQTT client ID (e.g. "transponder").
- `MqttClient * mqtt` = nullptr
MQTT client instance.
- `bool m_has_MQTT` = false
True once enough MQTT values have been received.
- `bool m_has_MQTT_gyro` = false
- `bool m_has_MQTT_accel` = false
- `bool m_has_MQTT_vsi` = false
- `bool m_has_MQTT_heading` = false
- `bool m_has_MQTT_airspeed` = false
- `bool m_has_MQTT_pressure` = false
- `Callbacks * callbacks` = nullptr
Global callback holder (must be created/released elsewhere).
- `QString imuData`
Last raw IMU ASCII payload.
- `bool Transponderstat` = false
True if transponder is connected and open.

- bool **Altitudestat** = false
For convenience on macOS (no USB check yet).
- bool **Radarstat** = false
True if radar device is connected.
- float **rPos** = 0.0f
Raw radar "position" / bearing.
- float **rSpeed** = 0.0f
Radar radial speed along beam.
- float **rDist** = 0.0f
Radar distance along beam.
- bool **IMUconnected** = false
True once IMU/INS is detected and streaming.
- bool **m_external** = true
True if using external (HW) IMU vs simulation.
- bool **m_imu_setup_done** = false
True once IMU initialization has completed.
- double **m_pmeasure_QNH** = -10000
Pressure-based altitude (feet), -10000 if invalid.
- bool **TransponderstatWithBarometer** = false
True if transponder has built-in barometer.
- QString **FromID**
- double **AccX** = 0.0
Acceleration X [m/s^2].
- double **AccY** = 0.0
Acceleration Y [m/s^2].
- double **AccZ** = 0.0
Acceleration Z [m/s^2].
- double **G** = Gfix
Local gravity constant.
- double **AsX** = 0.0
Angular speed X [deg/s or rad/s, depending on sensor].
- double **AsY** = 0.0
- double **AsZ** = 0.0
- double **AngleX** = 0.0
Roll angle [deg].
- double **AngleY** = 0.0
Pitch angle [deg].
- double **AngleZ** = 0.001
Yaw angle [deg].
- double **HX** = 0.0
Magnetometer X.
- double **HY** = 0.0
Magnetometer Y.
- double **HZ** = 0.0
Magnetometer Z.
- quint16 **VER** = 0.0
IMU firmware version or similar (from "VER" field).
- double **Temp** = -100.0
IMU temperature [°C].
- double **m_altitude** = -100.0
GPS altitude [feet] (geoid-compensated).

- double `m_latitude` = 0.0
GPS latitude [deg].
- double `m_longitude` = 0.0
GPS longitude [deg].
- double `m_gpsspeed` = 0.0
GPS speed [m/s].
- double `m_gpsbearing` = 0.0
GPS course [deg].
- double `m_vel_N` = 0.0
North velocity [m/s].
- double `m_vel_E` = 0.0
East velocity [m/s].
- double `m_vel_D` = 0.0
Down velocity [m/s].
- bool `m_vel_active` = false
- double `m_pressure_alt` = 0.0
Barometric altitude [feet].
- double `m_pressure` = 0.0
Barometric pressure [hPa].
- double `m_pressure_raw` = 0.0
Raw barometric pressure [hPa] before offsets.
- double `m_speed` = 0.0
Ground speed [km/h].
- double `Donwn_Speed` = 0.0
Vertical speed (down) [m/s].
- int `Orient` = 0
Orientation mode / sensor orientation index.
- bool `use_ins_only` = false

8.24.1 Detailed Description

Handles communication with:

- Transponder (USB serial)
- Radar (USB serial or simulated)
- IMU / INS (Bluetooth or USB serial)
- X-Plane via MQTT (simulated sensor data)

It also coordinates startup, periodically polls the transponder, and exposes decoded IMU / radar values as public members.

8.24.2 Member Typedef Documentation

8.24.2.1 Callbacks

```
typedef struct MyTcpSocket::Callbacks MyTcpSocket::Callbacks
```

8.24.3 Constructor & Destructor Documentation

8.24.3.1 MyTcpSocket()

```
MyTcpSocket::MyTcpSocket (
    QObject * parent = nullptr,
    QPlainTextEdit * s = nullptr,
    void(* retx )(void *, const char *, uint32_t) = nullptr,
    void(* rety )(void *, bool use_imu) = nullptr) [explicit]
```

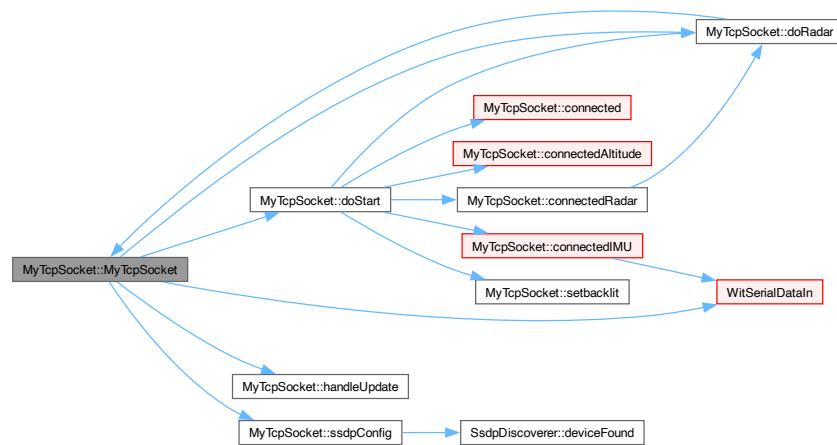
Construct a [MyTcpSocket](#).

Construct a [MyTcpSocket](#) object.

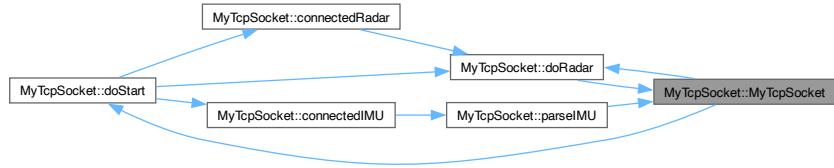
Parameters

<i>parent</i>	QObject parent for ownership.
<i>s</i>	Optional text widget used for logging / debug output.
<i>retx</i>	C-style callback invoked when transponder data is received.
<i>rety</i>	C-style callback used to report IMU connection status.
<i>parent</i>	QObject parent (Qt ownership).
<i>s</i>	Pointer to log / debug text output widget.
<i>retx</i>	Callback invoked by serial ports (transponder) when data arrives.
<i>rety</i>	Callback used to inform about IMU connection status.

Here is the call graph for this function:



Here is the caller graph for this function:



8.24.3.2 ~MyTcpSocket()

```
MyTcpSocket ::~MyTcpSocket ()
```

Destructor. Closes all serial ports.

8.24.4 Member Function Documentation

8.24.4.1 CallSwiftMemberFromC()

```
void MyTcpSocket ::CallSwiftMemberFromC (
    void * classPtr,
    void(* callback )(void *) ) [inline]
```

Example bridge to invoke Swift member functions from C code.

Stores class pointer + function pointer in callbacks and executes it via a lambda. Actual Swift bridging logic lives outside this file.

8.24.4.2 com_setup()

```
int MyTcpSocket ::com_setup (
    QSerialPort * com_port,
    QString sport)
```

Configure a QSerialPort with given port name.

Parameters

<i>com_port</i>	Port instance.
<i>sport</i>	System port name (e.g. "COM3").

Returns

0 on success, non-zero on error.

8.24.4.3 connected()

```
void MyTcpSocket::connected ()
```

Try to connect and initialize the transponder.

Try to connect transponder on the configured serial port.

Starts transponder polling timer on success.

On success:

- Sets Transponderstat = true
- Queries version and configuration
- Starts timerTRANS to periodically call [doTransponder\(\)](#)

Here is the call graph for this function:



Here is the caller graph for this function:



8.24.4.4 connectedAltitude()

```
void MyTcpSocket::connectedAltitude ()
```

Try to connect and initialize RADAR (Net or USB).

Try to connect transponder on the configured serial port.

Radar, with status dialogs along the way.

On success:

- Sets Transponderstat = true
- Queries version and configuration

- Starts timerTRANS to periodically call `doTransponder()`

Here is the call graph for this function:



Here is the caller graph for this function:



8.24.4.5 `connectedIMU()`

```
void MyTcpSocket::connectedIMU ()
```

Try to connect and initialize IMU (BT and/or USB) and Radar.

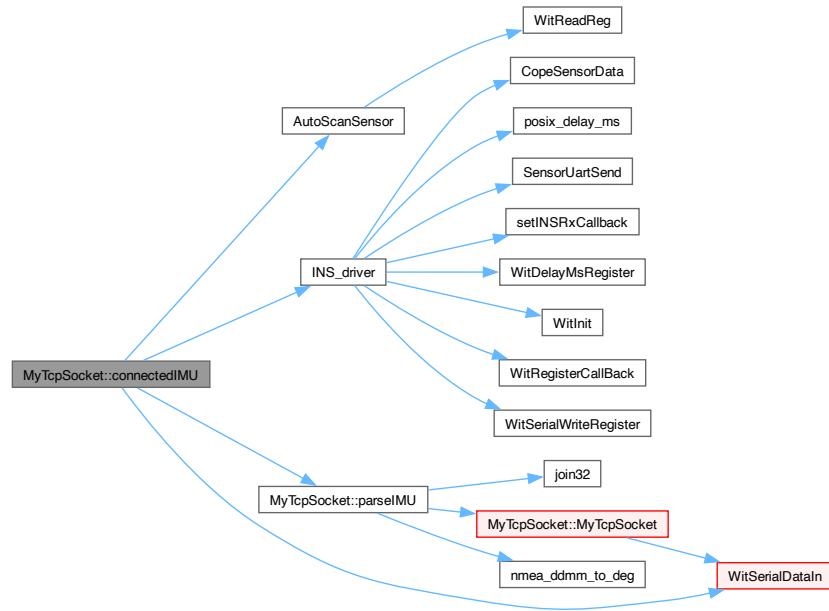
Try to connect to IMU/INS and Radar devices.

BT IMU → USB IMU → Radar, with status dialogs along the way.

Order:

1. Bluetooth IMU (WT901 BLE) if enabled.
2. Serial IMU WTGAHRS1/3.
3. Wlan IMU WTGAHRS1/3.

Here is the call graph for this function:



Here is the caller graph for this function:



8.24.4.6 connectedRadar()

```
void MyTcpSocket::connectedRadar ()
```

Try to connect and initialize RADAR (Net or USB).

Radar, with status dialogs along the way. Here is the call graph for this function:



Here is the caller graph for this function:



8.24.4.7 doConnect()

```
void MyTcpSocket::doConnect ()
```

Initiate TCP/UDP connection if used (currently unused/stub).

Left as API hook for potential future network functionality.

8.24.4.8 doRadar()

```
void MyTcpSocket::doRadar (
    void * parent,
    const char * data,
    uint32_t length) [static]
```

Static callback for Radar data (called from C driver).

Static callback for Radar serial data.

Parameters

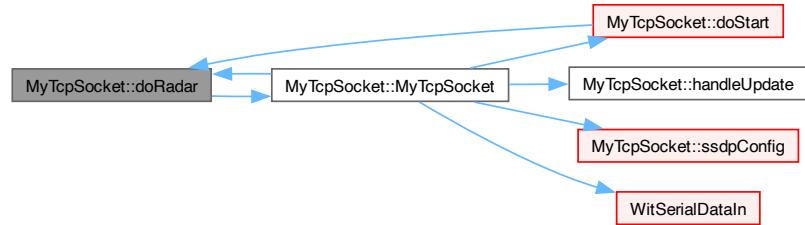
<i>parent</i>	Pointer back to MyTcpSocket instance.
<i>data</i>	Null-terminated ASCII payload.
<i>length</i>	Payload length in bytes.

Expected data format: "pos,radialSpeed,radialDist" Performs simple geometry using a fixed azimuth.

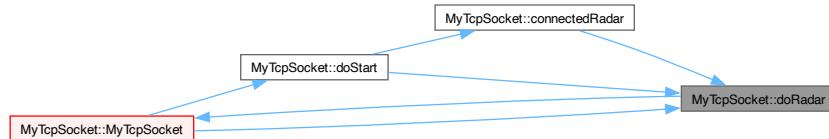
Parameters

<i>parent</i>	Pointer back to MyTcpSocket instance.
<i>data</i>	Null-terminated ASCII CSV.
<i>length</i>	Data length.

Here is the call graph for this function:



Here is the caller graph for this function:



8.24.4.9 doStart()

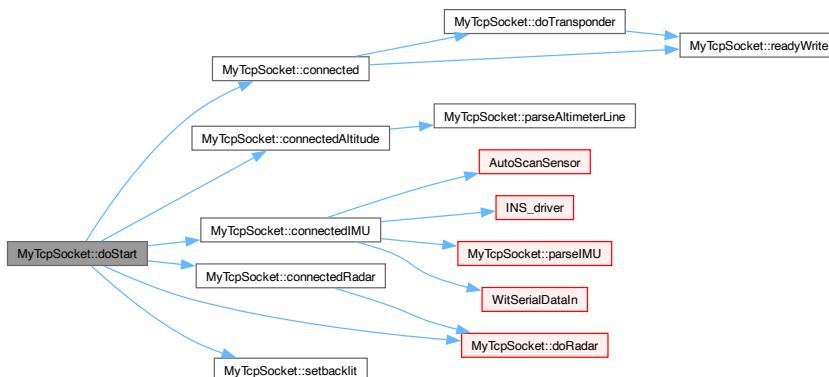
```
void MyTcpSocket::doStart ()
```

Startup state machine handler.

Progressive startup handler.

Runs through IMU → Transponder → Radar setup steps.

State machine driven by timerStart: state 0: Try connect IMU (BT / serial). state 1: Try connect transponder (USB). state 2: Keep monitoring / retry transponder. Also optionally generates radar/IMU simulation data when SIMULATE_RADAR is set. Here is the call graph for this function:



Here is the caller graph for this function:



8.24.4.10 doTransponder()

```
void MyTcpSocket::doTransponder ()
```

Periodic transponder polling / configuration state machine.

Periodically poll/configure the transponder.

Called from timerAlt every ~150 ms once transponder is connected.

Simple state-machine that:

- Asks for version, altitude, configuration, etc.
- Optionally updates altitude based on barometer (if present).

Called from timerAlt every ~150 ms. Here is the call graph for this function:



Here is the caller graph for this function:



8.24.4.11 handleUpdate()

```
void MyTcpSocket::handleUpdate (
    const std::string & ID,
    const std::string & invalue)
```

Process an incoming MQTT message from the X-Plane bridge.

Handle MQTT / X-Plane updates and map them to internal state.

Maps topics such as "xplane/roll", "xplane/ax", etc. into internal IMU state variables and sets MQTT presence flags.

Parameters

<i>ID</i>	Topic name (e.g. "xplane/roll").
<i>value</i>	Parsed float payload.

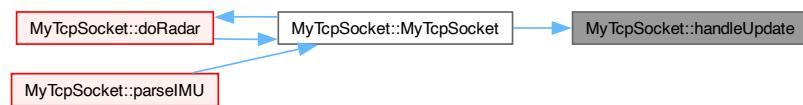
Topics currently supported (examples):

- xplane/ax, xplane/ay, xplane/az : accelerometer / gyro
- xplane/roll, xplane/pitch, xplane/yaw : attitude
- xplane/airspeed, xplane/climbRate : air data
- xplane/localPressure : barometric pressure

Parameters

<i>ID</i>	Topic string.
<i>value</i>	Parsed float payload.

Here is the caller graph for this function:



8.24.4.12 parseAltimeterLine()

```
void MyTcpSocket::parseAltimeterLine (
    const QString & line)
```

Here is the caller graph for this function:



8.24.4.13 parseIMU()

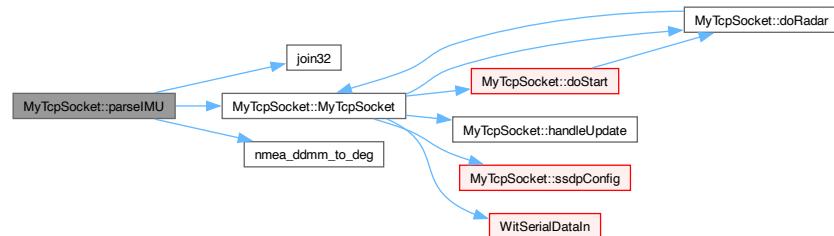
```
void MyTcpSocket::parseIMU (
    void * parent,
    uint32_t uiReg,
    uint16_t sReg[ ]) [static]
```

Static callback for IMU data (called from C driver).

Parameters

<i>parent</i>	Pointer back to MyTcpSocket instance.
<i>data</i>	Null-terminated ASCII payload.
<i>length</i>	Payload length in bytes.

Here is the call graph for this function:



Here is the caller graph for this function:



8.24.4.14 readyWrite()

```
void MyTcpSocket::readyWrite (
    char * data)
```

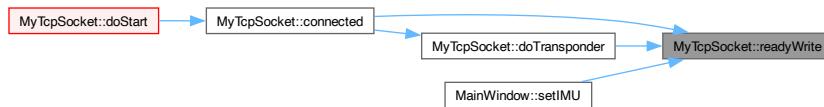
Send a raw ASCII command to the transponder, if connected.

Parameters

<i>data</i>	Null-terminated C string.
-------------	---------------------------

<i>data</i>	Null-terminated C string (command).
-------------	-------------------------------------

Here is the caller graph for this function:



8.24.4.15 sendMessage()

```
void MyTcpSocket::sendMessage (
    const QString & message)
```

Generic Qt signal for sending status / log messages to UI.

8.24.4.16 setbacklit()

```
void MyTcpSocket::setbacklit ()
```

Android: periodically bump external display backlight to max.

On Android, periodically set the external transponder display backlight.

No-op on non-Android platforms.

Uses Java class: com.hoho.android.usbserial.driver.TestClassTerje Here is the caller graph for this function:



8.24.4.17 setSerialPorts()

```
void MyTcpSocket::setSerialPorts (
    QString imu,
    QString transponder,
    QString radar)
```

Configure which devices to look for by "serial ID" or logical ID.

Configure which serial numbers / IDs are used.

On macOS this is typically the USB serial number mapped to /dev/tty device.

Parameters

<i>imu</i>	IMU/INS device identifier / serial number.
<i>transponder</i>	Transponder device identifier / serial number.
<i>radar</i>	Radar device identifier / serial number.
<i>imu</i>	IMU/INS device serial number or identifier.
<i>transponder</i>	Transponder device serial number or identifier.
<i>radar</i>	Radar device serial number or identifier.

8.24.4.18 ssdpConfig()

```
void MyTcpSocket::ssdpConfig ()
```

multicastConfig. Sends multicast...

Here is the call graph for this function:



Here is the caller graph for this function:



8.24.5 Field Documentation

8.24.5.1 _IMU_id

```
QString MyTcpSocket::_IMU_id
```

8.24.5.2 _radar_id

```
QString MyTcpSocket::_radar_id
```

8.24.5.3 _transponder_id

```
QString MyTcpSocket::_transponder_id
```

Device identifiers (serial numbers / logical IDs).

8.24.5.4 AccX

```
double MyTcpSocket::AccX = 0.0
```

Acceleration X [m/s²].

8.24.5.5 AccY

```
double MyTcpSocket::AccY = 0.0
```

Acceleration Y [m/s²].

8.24.5.6 AccZ

```
double MyTcpSocket::AccZ = 0.0
```

Acceleration Z [m/s²].

8.24.5.7 Altimeter_data

```
AltimeterData MyTcpSocket::Altimeter_data = {0,0,0,0}
```

8.24.5.8 Altitudestat

```
bool MyTcpSocket::Altitudestat = false
```

For convenience on macOS (no USB check yet).

8.24.5.9 AngleX

```
double MyTcpSocket::AngleX = 0.0
```

Roll angle [deg].

8.24.5.10 AngleY

```
double MyTcpSocket::AngleY = 0.0
```

Pitch angle [deg].

8.24.5.11 AngleZ

```
double MyTcpSocket::AngleZ = 0.001
```

Yaw angle [deg].

8.24.5.12 AsX

```
double MyTcpSocket::AsX = 0.0
```

Angular speed X [deg/s or rad/s, depending on sensor].

8.24.5.13 AsY

```
double MyTcpSocket::AsY = 0.0
```

8.24.5.14 AsZ

```
double MyTcpSocket::AsZ = 0.0
```

8.24.5.15 callbacks

```
Callbacks* MyTcpSocket::callbacks = nullptr
```

Global callback holder (must be created/released elsewhere).

8.24.5.16 CLIENT_ID

```
std::string MyTcpSocket::CLIENT_ID
```

MQTT client ID (e.g. "transponder").

8.24.5.17 disc

```
SsdpDiscoverer* MyTcpSocket::disc = nullptr
```

8.24.5.18 Donwn_Speed

```
double MyTcpSocket::Donwn_Speed = 0.0
```

Vertical speed (down) [m/s].

8.24.5.19 FromID

```
QString MyTcpSocket::FromID
```

8.24.5.20 G

```
double MyTcpSocket::G = Gfix
```

Local gravity constant.

8.24.5.21 HX

```
double MyTcpSocket::HX = 0.0
```

Magnetometer X.

8.24.5.22 HY

```
double MyTcpSocket::HY = 0.0
```

Magnetometer Y.

8.24.5.23 HZ

```
double MyTcpSocket::HZ = 0.0
```

Magnetometer Z.

8.24.5.24 IMUconnected

```
bool MyTcpSocket::IMUconnected = false
```

True once IMU/INS is detected and streaming.

8.24.5.25 imuData

```
QString MyTcpSocket::imuData
```

Last raw IMU ASCII payload.

8.24.5.26 lidar

```
QSerialPort* MyTcpSocket::lidar = nullptr
```

Optional lidar serial port (if used elsewhere).

8.24.5.27 m_address

```
QString MyTcpSocket::m_address = "239.255.0.1"
```

8.24.5.28 m_altitude

```
double MyTcpSocket::m_altitude = -100.0
```

GPS altitude [feet] (geoid-compensated).

8.24.5.29 m_external

```
bool MyTcpSocket::m_external = true
```

True if using external (HW) IMU vs simulation.

8.24.5.30 m_gpsbearing

```
double MyTcpSocket::m_gpsbearing = 0.0
```

GPS course [deg].

8.24.5.31 m_gpsspeed

```
double MyTcpSocket::m_gpsspeed = 0.0
```

GPS speed [m/s].

8.24.5.32 m_has_MQTT

```
bool MyTcpSocket::m_has_MQTT = false
```

True once enough MQTT values have been received.

8.24.5.33 m_has_MQTT_accel

```
bool MyTcpSocket::m_has_MQTT_accel = false
```

8.24.5.34 m_has_MQTT_airspeed

```
bool MyTcpSocket::m_has_MQTT_airspeed = false
```

8.24.5.35 m_has_MQTT_gyro

```
bool MyTcpSocket::m_has_MQTT_gyro = false
```

8.24.5.36 m_has_MQTT_heading

```
bool MyTcpSocket::m_has_MQTT_heading = false
```

8.24.5.37 m_has_MQTT_pressure

```
bool MyTcpSocket::m_has_MQTT_pressure = false
```

8.24.5.38 m_has_MQTT_vsi

```
bool MyTcpSocket::m_has_MQTT_vsi = false
```

8.24.5.39 m_imu_setup_done

```
bool MyTcpSocket::m_imu_setup_done = false
```

True once IMU initialization has completed.

8.24.5.40 m_latitude

```
double MyTcpSocket::m_latitude = 0.0
```

GPS latitude [deg].

8.24.5.41 m_longitude

```
double MyTcpSocket::m_longitude = 0.0
```

GPS longitude [deg].

8.24.5.42 m_pressure

```
double MyTcpSocket::m_pressure = 0.0
```

Barometric pressure [hPa].

8.24.5.43 m_pressure_alt

```
double MyTcpSocket::m_pressure_alt = 0.0
```

Barometric altitude [feet].

8.24.5.44 m_pressure_QNH

```
double MyTcpSocket::m_pressure_QNH = -10000
```

Pressure-based altitude (feet), -10000 if invalid.

8.24.5.45 m_pressure_raw

```
double MyTcpSocket::m_pressure_raw = 0.0
```

Raw barometric pressure [hPa] before offsets.

8.24.5.46 m_speed

```
double MyTcpSocket::m_speed = 0.0
```

Ground speed [km/h].

8.24.5.47 m_vel_active

```
bool MyTcpSocket::m_vel_active = false
```

8.24.5.48 m_vel_D

```
double MyTcpSocket::m_vel_D = 0.0
```

Down velocity [m/s].

8.24.5.49 m_vel_E

```
double MyTcpSocket::m_vel_E = 0.0
```

East velocity [m/s].

8.24.5.50 m_vel_N

```
double MyTcpSocket::m_vel_N = 0.0
```

North velocity [m/s].

8.24.5.51 MCAST_PORT

```
quint16 MyTcpSocket::MCAST_PORT = 4210
```

8.24.5.52 mqtt

```
MqttClient* MyTcpSocket::mqtt = nullptr
```

MQTT client instance.

8.24.5.53 Orient

```
int MyTcpSocket::Orient = 0
```

Orientation mode / sensor orientation index.

8.24.5.54 port

```
QSerialPort* MyTcpSocket::port = nullptr
```

Generic serial port (unused here; legacy).

8.24.5.55 Radarstat

```
bool MyTcpSocket::Radarstat = false
```

True if radar device is connected.

8.24.5.56 rDist

```
float MyTcpSocket::rDist = 0.0f
```

Radar distance along beam.

8.24.5.57 ret_imu

```
void(* MyTcpSocket::ret_imu) (void *, bool use_imu) = nullptr
```

Callback to report IMU usage/availability to external C code.

8.24.5.58 ret_transponder

```
void(* MyTcpSocket::ret_transponder) (void *, const char *data, uint32_t size) = nullptr
```

Callback invoked when serial data arrives from transponder.

8.24.5.59 rPos

```
float MyTcpSocket::rPos = 0.0f
```

Raw radar "position" / bearing.

8.24.5.60 rSpeed

```
float MyTcpSocket::rSpeed = 0.0f
```

Radar radial speed along beam.

8.24.5.61 serialport

```
QList<QSerialPortInfo> MyTcpSocket::serialport
```

Cached list of available serial ports.

8.24.5.62 SERVER_ADDRESS

```
std::string MyTcpSocket::SERVER_ADDRESS
```

MQTT broker address (e.g. "tcp://localhost:1883").

8.24.5.63 sport

```
QString MyTcpSocket::sport
```

Current serial port name (used by some legacy code).

8.24.5.64 Temp

```
double MyTcpSocket::Temp = -100.0
```

IMU temperature [°C].

8.24.5.65 text

```
QPlainTextEdit* MyTcpSocket::text = nullptr
```

Optional text log widget.

8.24.5.66 Transponderstat

```
bool MyTcpSocket::Transponderstat = false
```

True if transponder is connected and open.

8.24.5.67 TransponderstatWithBarometer

```
bool MyTcpSocket::TransponderstatWithBarometer = false
```

True if transponder has built-in barometer.

8.24.5.68 use_ins_only

```
bool MyTcpSocket::use_ins_only = false
```

8.24.5.69 VER

```
quint16 MyTcpSocket::VER = 0.0
```

IMU firmware version or similar (from "VER" field).

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

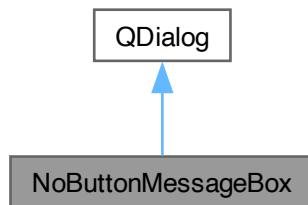
- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header/generic/[mytcpsocket.h](#)
- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/generic/[mytcpsocket.cpp](#)

8.25 NoButtonMessageBox Class Reference

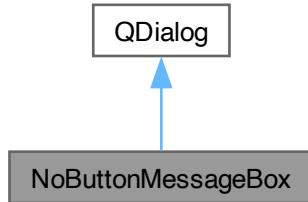
Small frameless dialog for transient status messages.

```
#include <mytcpsocket.h>
```

Inheritance diagram for NoButtonMessageBox:



Collaboration diagram for NoButtonMessageBox:



Public Member Functions

- [NoButtonMessageBox](#) (const QString &message, QWidget *parent=nullptr)
- void [setText](#) (const QString &text)
Update message text at runtime.

8.25.1 Detailed Description

Small frameless dialog for transient status messages.

A simple message box with no buttons, used to show progress / info like "Looking for USB Transponder!" that auto-hides after a delay.

8.25.2 Constructor & Destructor Documentation

8.25.2.1 NoButtonMessageBox()

```
NoButtonMessageBox::NoButtonMessageBox (const QString & message, QWidget * parent = nullptr) [inline], [explicit]
```

8.25.3 Member Function Documentation

8.25.3.1 setText()

```
void NoButtonMessageBox::setText (const QString & text) [inline]
```

Update message text at runtime.

Parameters

<i>text</i>	New text to show.
-------------	-------------------

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

- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header/generic/mytcpsocket.h

8.26 PortEntry Struct Reference

Convenience record for describing a serial port (used on macOS).

```
#include <mytcpsocket.h>
```

Data Fields

- **QString serial**
Device serial number (e.g. "4150323833373205").
- **QString portName**
User-facing port name (e.g. "COM5" or "cu.usbmodem1301").
- **QString systemLocation**
System path (e.g. "/dev/cu.usbmodem1301").
- **QString description**
Device description.
- **QString manufacturer**
Manufacturer string.
- **quint16 vendorId = 0**
USB vendor ID, if available.
- **quint16 productId = 0**
USB product ID, if available.

8.26.1 Detailed Description

Convenience record for describing a serial port (used on macOS).

8.26.2 Field Documentation

8.26.2.1 description

```
QString PortEntry::description
```

Device description.

8.26.2.2 manufacturer

```
QString PortEntry::manufacturer
```

Manufacturer string.

8.26.2.3 portName

```
QString PortEntry::portName
```

User-facing port name (e.g. "COM5" or "cu.usbmodem1301").

8.26.2.4 productId

```
quint16 PortEntry::productId = 0
```

USB product ID, if available.

8.26.2.5 serial

```
QString PortEntry::serial
```

Device serial number (e.g. "4150323833373205").

8.26.2.6 systemLocation

```
QString PortEntry::systemLocation
```

System path (e.g. "/dev/cu.usbmodem1301").

8.26.2.7 vendorId

```
quint16 PortEntry::vendorId = 0
```

USB vendor ID, if available.

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

- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header/generic/mytcpsocket.h

8.27 QuantumCircuit Class Reference

```
#include <MicroQiskitCpp.h>
```

Public Member Functions

- `QuantumCircuit ()`
- `QuantumCircuit (int n, int m=0)`
- `void set_registers (int n, int m=0)`
- `void add (QuantumCircuit qc2)`
- `void initialize (vector< double > p)`
- `void x (int q)`
- `void rx (double theta, int q)`
- `void h (int q)`
- `void cx (int s, int t)`
- `void ch (int s, int t)`
- `void crx (double theta, int s, int t)`
- `void measure (int q, int b)`
- `void rz (double theta, int q)`
- `void ry (double theta, int q)`
- `void z (int q)`
- `void y (int q)`
- `bool has_measurements ()`

Data Fields

- `int nQubits`
- `int nBits`
- `vector< vector< string > > data`

8.27.1 Constructor & Destructor Documentation

8.27.1.1 QuantumCircuit() [1/2]

`QuantumCircuit::QuantumCircuit ()` [inline]

Here is the caller graph for this function:



8.27.1.2 QuantumCircuit() [2/2]

```
QuantumCircuit::QuantumCircuit (
    int n,
    int m = 0)  [inline]
```

Here is the call graph for this function:



8.27.2 Member Function Documentation

8.27.2.1 add()

```
void QuantumCircuit::add (
    QuantumCircuit qc2)  [inline]
```

Here is the call graph for this function:



Here is the caller graph for this function:



8.27.2.2 ch()

```
void QuantumCircuit::ch (
    int s,
    int t) [inline]
```

8.27.2.3 crx()

```
void QuantumCircuit::crx (
    double theta,
    int s,
    int t) [inline]
```

8.27.2.4 cx()

```
void QuantumCircuit::cx (
    int s,
    int t) [inline]
```

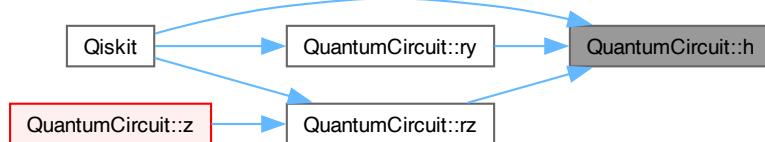
Here is the caller graph for this function:



8.27.2.5 h()

```
void QuantumCircuit::h (
    int q) [inline]
```

Here is the caller graph for this function:



8.27.2.6 has_measurements()

```
bool QuantumCircuit::has_measurements () [inline]
```

8.27.2.7 initialize()

```
void QuantumCircuit::initialize (
    vector< double > p) [inline]
```

8.27.2.8 measure()

```
void QuantumCircuit::measure (
    int q,
    int b) [inline]
```

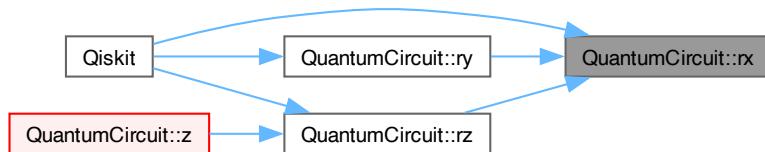
Here is the caller graph for this function:



8.27.2.9 rx()

```
void QuantumCircuit::rx (
    double theta,
    int q) [inline]
```

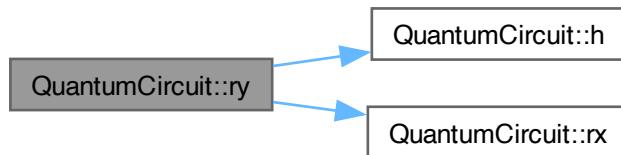
Here is the caller graph for this function:



8.27.2.10 ry()

```
void QuantumCircuit::ry (
    double theta,
    int q) [inline]
```

Here is the call graph for this function:



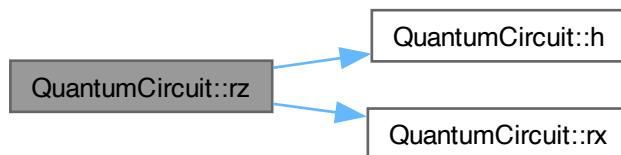
Here is the caller graph for this function:



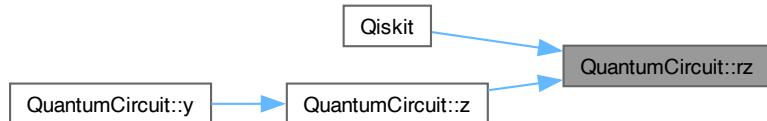
8.27.2.11 rz()

```
void QuantumCircuit::rz (
    double theta,
    int q) [inline]
```

Here is the call graph for this function:



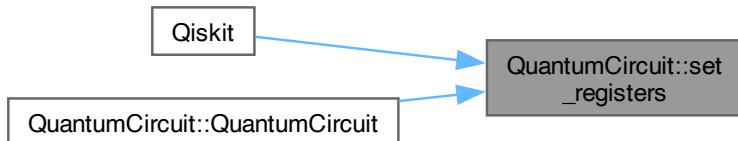
Here is the caller graph for this function:



8.27.2.12 set_registers()

```
void QuantumCircuit::set_registers (
    int n,
    int m = 0) [inline]
```

Here is the caller graph for this function:



8.27.2.13 x()

```
void QuantumCircuit::x (
    int q) [inline]
```

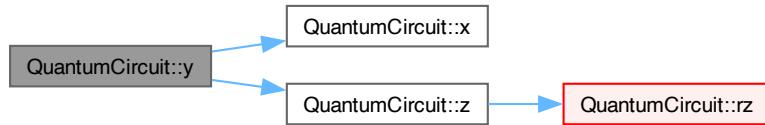
Here is the caller graph for this function:



8.27.2.14 y()

```
void QuantumCircuit::y (
    int q) [inline]
```

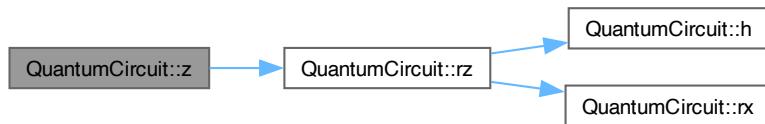
Here is the call graph for this function:



8.27.2.15 z()

```
void QuantumCircuit::z (
    int q) [inline]
```

Here is the call graph for this function:



Here is the caller graph for this function:



8.27.3 Field Documentation

8.27.3.1 data

```
vector<vector<string>> QuantumCircuit::data
```

8.27.3.2 nBits

```
int QuantumCircuit::nBits
```

8.27.3.3 nQubits

```
int QuantumCircuit::nQubits
```

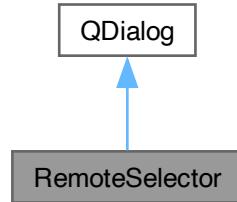
The documentation for this class was generated from the following file:

- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header/generic/[MicroQiskitCpp.h](#)

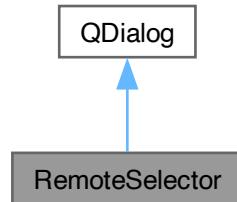
8.28 RemoteSelector Class Reference

```
#include <remoteselector.h>
```

Inheritance diagram for RemoteSelector:



Collaboration diagram for RemoteSelector:



Public Member Functions

- `RemoteSelector (const QBluetoothAddress &localAdapter, QWidget *parent=nullptr)`
- `~RemoteSelector ()`
- `void startDiscovery (const QBluetoothUuid &uuid)`
- `void stopDiscovery ()`
- `QBluetoothServiceInfo service () const`

8.28.1 Constructor & Destructor Documentation

8.28.1.1 RemoteSelector()

```
QT_USE_NAMESPACE RemoteSelector::RemoteSelector (
    const QBluetoothAddress & localAdapter,
    QWidget * parent = nullptr) [explicit]
```

Here is the call graph for this function:



Here is the caller graph for this function:



8.28.1.2 ~RemoteSelector()

```
RemoteSelector::~RemoteSelector ()
```

8.28.2 Member Function Documentation

8.28.2.1 service()

```
QBluetoothServiceInfo RemoteSelector::service () const
```

8.28.2.2 startDiscovery()

```
void RemoteSelector::startDiscovery (
    const QBluetoothUuid & uuid)
```

8.28.2.3 stopDiscovery()

```
void RemoteSelector::stopDiscovery ()
```

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

- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header/generic/[remoteselector.h](#)
- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/generic/[remoteselector.cpp](#)

8.29 SensorService Struct Reference

```
#include <ssdp.h>
```

Data Fields

- const char * [name](#)
- const char * [st](#)
- const char * [uuid](#)

8.29.1 Field Documentation

8.29.1.1 name

```
const char* SensorService::name
```

8.29.1.2 st

```
const char* SensorService::st
```

8.29.1.3 uuid

```
const char* SensorService::uuid
```

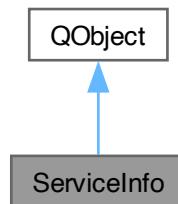
The documentation for this struct was generated from the following file:

- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header/generic/ssdp.h

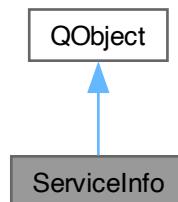
8.30 ServiceInfo Class Reference

```
#include <serviceinfo.h>
```

Inheritance diagram for ServiceInfo:



Collaboration diagram for ServiceInfo:



Public Member Functions

- `ServiceInfo ()=default`
- `ServiceInfo (QLowEnergyService *service)`
- `QLowEnergyService * service () const`
- `QString getUuid () const`
- `QString getName () const`
- `QString getType () const`
- `void serviceChanged ()`

8.30.1 Constructor & Destructor Documentation

8.30.1.1 ServiceInfo() [1/2]

```
ServiceInfo::ServiceInfo () [default]
```

Here is the call graph for this function:



8.30.1.2 ServiceInfo() [2/2]

```
ServiceInfo::ServiceInfo (
    QLowEnergyService * service)
```

Here is the call graph for this function:



8.30.2 Member Function Documentation

8.30.2.1 getName()

```
QString ServiceInfo::getName () const
```

8.30.2.2 getType()

```
QString ServiceInfo::getType () const
```

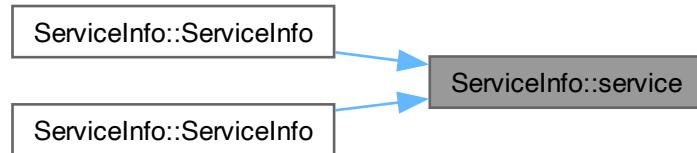
8.30.2.3 getUuid()

```
QString ServiceInfo::getUuid () const
```

8.30.2.4 service()

```
QLowEnergyService * ServiceInfo::service () const
```

Here is the caller graph for this function:



8.30.2.5 serviceChanged()

```
void ServiceInfo::serviceChanged ()
```

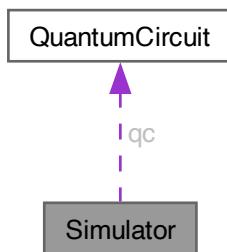
The documentation for this class was generated from the following files:

- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header/generic/[serviceinfo.h](#)
- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/generic/[serviceinfo.cpp](#)

8.31 Simulator Class Reference

```
#include <MicroQiskitCpp.h>
```

Collaboration diagram for Simulator:



Public Member Functions

- `Simulator (QuantumCircuit qc_in, int shots_in=1024)`
- `vector< complex< double >> get_statevector ()`
- `vector< string > get_memory ()`
- `map< string, int > get_counts ()`
- `string get_qiskit ()`
- `string get_qasm ()`

Data Fields

- `QuantumCircuit qc`
- `int shots`

8.31.1 Constructor & Destructor Documentation

8.31.1.1 Simulator()

```
Simulator::Simulator (
    QuantumCircuit qc_in,
    int shots_in = 1024) [inline]
```

8.31.2 Member Function Documentation

8.31.2.1 get_counts()

```
map< string, int > Simulator::get_counts () [inline]
```

Here is the call graph for this function:



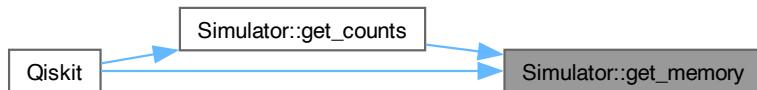
Here is the caller graph for this function:



8.31.2.2 get_memory()

```
vector< string > Simulator::get_memory () [inline]
```

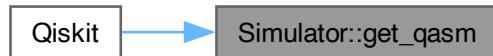
Here is the caller graph for this function:



8.31.2.3 get_qasm()

```
string Simulator::get_qasm () [inline]
```

Here is the caller graph for this function:



8.31.2.4 get_qiskit()

```
string Simulator::get_qiskit () [inline]
```

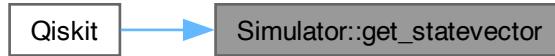
Here is the caller graph for this function:



8.31.2.5 get_statevector()

```
vector< complex< double > > Simulator::get_statevector () [inline]
```

Here is the caller graph for this function:



8.31.3 Field Documentation

8.31.3.1 qc

```
QuantumCircuit Simulator::qc
```

8.31.3.2 shots

```
int Simulator::shots
```

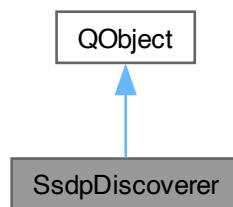
The documentation for this class was generated from the following file:

- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header/generic/[MicroQiskitCpp.h](#)

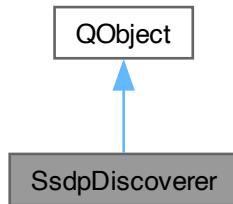
8.32 SsdpDiscoverer Class Reference

```
#include <ssdp.h>
```

Inheritance diagram for SsdpDiscoverer:



Collaboration diagram for SsdpDiscoverer:



Public Member Functions

- [SsdpDiscoverer](#) (QObject *parent=nullptr)
SSDP (Simple Service Discovery Protocol) helper class.
- void [startDiscovery](#) (int durationMs=2000)
Sends an SSDP M-SEARCH discovery request.
- void [deviceFound](#) (const QHostAddress &addr, quint16 port, const QString &st)

8.32.1 Constructor & Destructor Documentation

8.32.1.1 SsdpDiscoverer()

```
SsdpDiscoverer::SsdpDiscoverer (
    QObject * parent = nullptr) [explicit]
```

SSDP (Simple Service Discovery Protocol) helper class.

This class provides basic SSDP discovery support using UDP multicast. It can listen for SSDP NOTIFY messages and process incoming discovery traffic.

Responsibilities:

- Managing a UDP socket bound to the SSDP multicast group
- Receiving and parsing SSDP datagrams

Non-responsibilities:

- Parsing device description XML
- Managing device lifetimes
- Thread synchronization

Threading:

- Must be used from the thread it is created in

Typical usage:

- Create an instance
- Start listening for SSDP traffic
- React to parsed SSDP messages

8.32.2 Member Function Documentation

8.32.2.1 deviceFound()

```
void SsdpDiscoverer::deviceFound (
    const QHostAddress & addr,
    quint16 port,
    const QString & st)
```

Here is the caller graph for this function:



8.32.2.2 startDiscovery()

```
void SsdpDiscoverer::startDiscovery (
    int durationMs = 2000)
```

Sends an SSDP M-SEARCH discovery request.

Constructs and transmits a single SSDP M-SEARCH request to the standard SSDP multicast group in order to actively discover compatible devices or services on the local network.

The request uses the following SSDP headers:

- HOST : SSDP multicast address and port
- MAN : "ssdp:discover" (required for discovery requests)
- MX : Maximum wait time (in seconds) for responses
- ST : Search target (custom service type)

Responses, if any, are delivered asynchronously via the UDP socket and processed elsewhere in the class.

Parameters

<i>durationMs</i>	Requested discovery duration in milliseconds. Currently unused by this implementation; discovery consists of sending a single M-SEARCH request.
-------------------	---

Note

This function does not block and does not wait for responses.

A failure to send the datagram is reported via `qWarning()`.

See also`onReadyRead()`

Here is the call graph for this function:



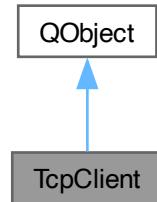
The documentation for this class was generated from the following files:

- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header/generic/[ssdp.h](#)
- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/generic/[ssdp.cpp](#)

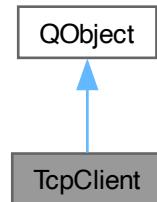
8.33 TcpClient Class Reference

```
#include <tcpclient.h>
```

Inheritance diagram for TcpClient:



Collaboration diagram for TcpClient:



Public Member Functions

- `TcpClient (QObject *parent=nullptr)`
- `void connectTo (const QHostAddress &address, quint16 port=23)`
- `void disconnectFrom ()`
- `void sendData (const QByteArray &data)`
- `void connected ()`
- `void disconnected ()`
- `void errorOccurred (const QString &error)`
- `void dataReceived (const QByteArray &data)`

8.33.1 Constructor & Destructor Documentation

8.33.1.1 `TcpClient()`

```
TcpClient::TcpClient (
    QObject * parent = nullptr) [explicit]
```

8.33.2 Member Function Documentation

8.33.2.1 `connected()`

```
void TcpClient::connected ()
```

8.33.2.2 `connectTo()`

```
void TcpClient::connectTo (
    const QHostAddress & address,
    quint16 port = 23)
```

8.33.2.3 `dataReceived()`

```
void TcpClient::dataReceived (
    const QByteArray & data)
```

8.33.2.4 `disconnected()`

```
void TcpClient::disconnected ()
```

8.33.2.5 `disconnectFrom()`

```
void TcpClient::disconnectFrom ()
```

8.33.2.6 `errorOccurred()`

```
void TcpClient::errorOccurred (
    const QString & error)
```

8.33.2.7 `sendData()`

```
void TcpClient::sendData (
    const QByteArray & data)
```

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

- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header/generic/[tcpclient.h](#)
- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/generic/[tcpclient.cpp](#)

8.34 TrackPoint Struct Reference

```
#include <gpx_parse.h>
```

Data Fields

- double `latitude`
- double `longitude`
- double `elevation`
- double `dt`
- QDateTime `time`
- double `speed` = 0.0

8.34.1 Field Documentation

8.34.1.1 `dt`

```
double TrackPoint::dt
```

8.34.1.2 `elevation`

```
double TrackPoint::elevation
```

8.34.1.3 `latitude`

```
double TrackPoint::latitude
```

8.34.1.4 longitude

```
double TrackPoint::longitude
```

8.34.1.5 speed

```
double TrackPoint::speed = 0.0
```

8.34.1.6 time

```
QDateTime TrackPoint::time
```

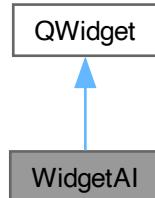
The documentation for this struct was generated from the following file:

- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header/driver/[gpx_parse.h](#)

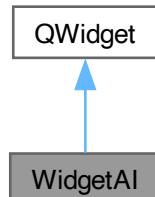
8.35 WidgetAI Class Reference

```
#include <WidgetAI.h>
```

Inheritance diagram for WidgetAI:



Collaboration diagram for WidgetAI:



Public Member Functions

- `WidgetAI (QWidget *parent=Q_NULLPTR)`
- `~WidgetAI ()`
- `void redraw ()`
- `void reinit ()`
- `void setRoll (double roll)`
- `void setPitch (double pitch)`

8.35.1 Constructor & Destructor Documentation

8.35.1.1 WidgetAI()

```
WidgetAI::WidgetAI (
    QWidget * parent = Q_NULLPTR) [explicit]
```

Here is the call graph for this function:



Here is the caller graph for this function:



8.35.1.2 ~WidgetAI()

```
WidgetAI::~WidgetAI ()
```

8.35.2 Member Function Documentation

8.35.2.1 `redraw()`

```
void WidgetAI::redraw () [inline]
```

8.35.2.2 `reinit()`

```
void WidgetAI::reinit () [inline]
```

8.35.2.3 `setPitch()`

```
void WidgetAI::setPitch (
    double pitch) [inline]
```

8.35.2.4 `setRoll()`

```
void WidgetAI::setRoll (
    double roll) [inline]
```

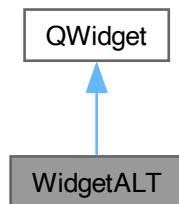
The documentation for this class was generated from the following files:

- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/WidgetAI.h
- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/WidgetAI.cpp

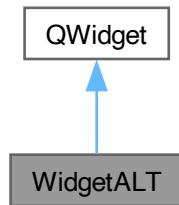
8.36 WidgetALT Class Reference

```
#include <WidgetALT.h>
```

Inheritance diagram for WidgetALT:



Collaboration diagram for WidgetALT:



Public Member Functions

- `WidgetALT (QWidget *parent=Q_NULLPTR)`
- `~WidgetALT ()`
- `void redraw ()`
- `void reinit ()`
- `void setAltitude (double altitude)`
- `void setPressure (double pressure)`

8.36.1 Constructor & Destructor Documentation

8.36.1.1 WidgetALT()

```
WidgetALT::WidgetALT (
    QWidget * parent = Q_NULLPTR) [explicit]
```

Here is the call graph for this function:



Here is the caller graph for this function:



8.36.1.2 ~WidgetALT()

```
WidgetALT::~WidgetALT ()
```

8.36.2 Member Function Documentation

8.36.2.1 redraw()

```
void WidgetALT::redraw () [inline]
```

8.36.2.2 reinit()

```
void WidgetALT::reinit () [inline]
```

8.36.2.3 setAltitude()

```
void WidgetALT::setAltitude (
    double altitude) [inline]
```

8.36.2.4 setPressure()

```
void WidgetALT::setPressure (
    double pressure) [inline]
```

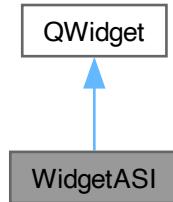
The documentation for this class was generated from the following files:

- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/WidgetALT.h
- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/WidgetALT.cpp

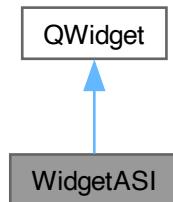
8.37 WidgetASI Class Reference

```
#include <WidgetASI.h>
```

Inheritance diagram for WidgetASI:



Collaboration diagram for WidgetASI:



Public Member Functions

- [WidgetASI \(QWidget *parent=Q_NULLPTR\)](#)
- [~WidgetASI \(\)](#)
- [void redraw \(\)](#)
- [void reinit \(\)](#)
- [void setAirspeed \(double airspeed\)](#)

8.37.1 Constructor & Destructor Documentation

8.37.1.1 WidgetASI()

```
WidgetASI::WidgetASI (
    QWidget * parent = Q_NULLPTR) [explicit]
```

Here is the call graph for this function:



Here is the caller graph for this function:



8.37.1.2 ~WidgetASI()

```
WidgetASI::~WidgetASI ()
```

8.37.2 Member Function Documentation

8.37.2.1 redraw()

```
void WidgetASI::redraw () [inline]
```

8.37.2.2 reinit()

```
void WidgetASI::reinit () [inline]
```

8.37.2.3 setAirspeed()

```
void WidgetASI::setAirspeed (
    double airspeed) [inline]
```

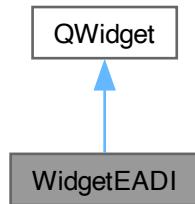
The documentation for this class was generated from the following files:

- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/WidgetASI.h
- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/WidgetASI.cpp

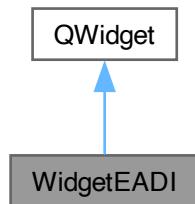
8.38 WidgetEADI Class Reference

```
#include <WidgetEADI.h>
```

Inheritance diagram for WidgetEADI:



Collaboration diagram for WidgetEADI:



Public Member Functions

- [WidgetEADI \(QWidget *parent=Q_NULLPTR\)](#)
- virtual [~WidgetEADI \(\)](#)
- void [reinit \(\)](#)
- void [redraw \(\)](#)
- void [setFltMode \(qfi_EADI::FltMode fltMode\)](#)
- void [setSpdMode \(qfi_EADI::SpdMode spdMode\)](#)
- void [setLNAV \(qfi_EADI::LNAV lnav\)](#)
- void [setVNAV \(qfi_EADI::VNAV vnav\)](#)
- void [setRoll \(double roll\)](#)
- void [setPitch \(double pitch\)](#)
- void [setFPM \(double aoa, double sideslip\)](#)
- void [setSlipSkid \(double slipSkid\)](#)
- void [setTurnRate \(double turnRate\)](#)
- void [setDots \(double dotH, double dotV, bool visibleH, bool visibleV\)](#)

- void `setFD` (double roll, double pitch, bool visible=true)
- void `setStall` (bool stall)
- void `setAltitude` (double altitude)
- void `setPressure` (double pressure, qfi_EADI::PressureMode pressMode)
- void `setAirspeed` (double airspeed)
- void `setMachNo` (double machNo)
- void `setHeading` (double heading)
- void `setClimbRate` (double climbRate)
- void `setAirspeedSel` (double airspeed)
- void `setAltitudeSel` (double altitude)
- void `setHeadingSel` (double heading)
- void `setVfe` (double vfe)
- void `setVne` (double vne)

8.38.1 Constructor & Destructor Documentation

8.38.1.1 WidgetEADI()

```
WidgetEADI::WidgetEADI (
    QWidget * parent = Q_NULLPTR) [explicit]
```

Here is the call graph for this function:



Here is the caller graph for this function:



8.38.1.2 ~WidgetEADI()

```
WidgetEADI::~WidgetEADI () [virtual]
```

8.38.2 Member Function Documentation

8.38.2.1 **redraw()**

```
void WidgetEADI::redraw () [inline]
```

8.38.2.2 **reinit()**

```
void WidgetEADI::reinit () [inline]
```

8.38.2.3 **setAirspeed()**

```
void WidgetEADI::setAirspeed (
    double airspeed) [inline]
```

8.38.2.4 **setAirspeedSel()**

```
void WidgetEADI::setAirspeedSel (
    double airspeed) [inline]
```

Parameters

<i>airspeed</i>	(dimensionless numeric value)
-----------------	-------------------------------

8.38.2.5 **setAltitude()**

```
void WidgetEADI::setAltitude (
    double altitude) [inline]
```

8.38.2.6 **setAltitudeSel()**

```
void WidgetEADI::setAltitudeSel (
    double altitude) [inline]
```

Parameters

<i>altitude</i>	(dimensionless numeric value)
-----------------	-------------------------------

8.38.2.7 **setClimbRate()**

```
void WidgetEADI::setClimbRate (
    double climbRate) [inline]
```

8.38.2.8 setDots()

```
void WidgetEADI::setDots (
    double dotH,
    double dotV,
    bool visibleH,
    bool visibleV) [inline]
```

Parameters

<i>normalized</i>	horizontal deviation dot position (range from -1.0 to 1.0)
<i>normalized</i>	vertical deviation dot position (range from -1.0 to 1.0)
<i>deviation</i>	horizontal dot visibility
<i>deviation</i>	vertical dot visibility

8.38.2.9 setFD()

```
void WidgetEADI::setFD (
    double roll,
    double pitch,
    bool visible = true) [inline]
```

Parameters

<i>FD</i>	roll angle [deg]
<i>FD</i>	pitch angle [deg]
<i>FD</i>	visibility

8.38.2.10 setFltMode()

```
void WidgetEADI::setFltMode (
    qfi_EADI::FltMode fltMode) [inline]
```

8.38.2.11 setFPM()

```
void WidgetEADI::setFPM (
    double aoa,
    double sideslip) [inline]
```

8.38.2.12 setHeading()

```
void WidgetEADI::setHeading (
    double heading) [inline]
```

8.38.2.13 setHeadingSel()

```
void WidgetEADI::setHeadingSel (
    double heading) [inline]
```

Parameters

<i>heading</i>	[deg]
----------------	-------

8.38.2.14 setLNAV()

```
void WidgetEADI::setLNAV (
    qfi_EADI::LNAV lnav) [inline]
```

8.38.2.15 setMachNo()

```
void WidgetEADI::setMachNo (
    double machNo) [inline]
```

8.38.2.16 setPitch()

```
void WidgetEADI::setPitch (
    double pitch) [inline]
```

8.38.2.17 setPressure()

```
void WidgetEADI::setPressure (
    double pressure,
    qfi_EADI::PressureMode pressMode) [inline]
```

8.38.2.18 setRoll()

```
void WidgetEADI::setRoll (
    double roll) [inline]
```

8.38.2.19 setSlipSkid()

```
void WidgetEADI::setSlipSkid (
    double slipSkid) [inline]
```

8.38.2.20 setSpdMode()

```
void WidgetEADI::setSpdMode (
    qfi_EADI::SpdMode spdMode) [inline]
```

8.38.2.21 setStall()

```
void WidgetEADI::setStall (
    bool stall) [inline]
```

Parameters

<i>stall</i>	<i>flag</i>
--------------	-------------

8.38.2.22 setTurnRate()

```
void WidgetEADI::setTurnRate (
    double turnRate) [inline]
```

Parameters

<i>normalized</i>	turn rate (range from -1.0 to 1.0), hash marks positions are set to be -0.5 and 0.5
-------------------	---

8.38.2.23 setVfe()

```
void WidgetEADI::setVfe (
    double vfe) [inline]
```

8.38.2.24 setVNAV()

```
void WidgetEADI::setVNAV (
    qfi_EADI::VNAV vnav) [inline]
```

8.38.2.25 setVne()

```
void WidgetEADI::setVne (
    double vne) [inline]
```

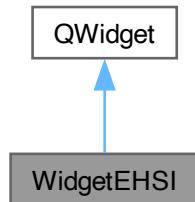
The documentation for this class was generated from the following files:

- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/WidgetEADI.h
- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/WidgetEADI.cpp

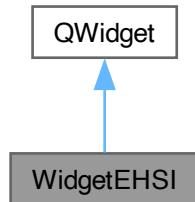
8.39 WidgetEHSI Class Reference

```
#include <WidgetEHSI.h>
```

Inheritance diagram for WidgetEHSI:



Collaboration diagram for WidgetEHSI:



Public Member Functions

- [WidgetEHSI](#) (QWidget *parent=Q_NULLPTR)
- virtual [~WidgetEHSI](#) ()
- void [reinit](#) ()
- void [redraw](#) ()
- void [setHeading](#) (double heading)
- void [setCourse](#) (double course)
- void [setBearing](#) (double bearing, bool visible=false)
- void [setDeviation](#) (double deviation, qfi_EHSI::CDI cdi)
- void [setDistance](#) (double distance, bool visible=false)
- void [setHeadingSel](#) (double headingBug)

8.39.1 Constructor & Destructor Documentation

8.39.1.1 WidgetEHSI()

```
WidgetEHSI::WidgetEHSI ( QWidget * parent = Q_NULLPTR) [explicit]
```

Here is the call graph for this function:



Here is the caller graph for this function:



8.39.1.2 ~WidgetEHSI()

```
WidgetEHSI::~WidgetEHSI () [virtual]
```

8.39.2 Member Function Documentation

8.39.2.1 redraw()

```
void WidgetEHSI::redraw () [inline]
```

8.39.2.2 reinit()

```
void WidgetEHSI::reinit () [inline]
```

8.39.2.3 setBearing()

```
void WidgetEHSI::setBearing (
    double bearing,
    bool visible = false) [inline]
```

8.39.2.4 setCourse()

```
void WidgetEHSI::setCourse (
    double course) [inline]
```

8.39.2.5 setDeviation()

```
void WidgetEHSI::setDeviation (
    double deviation,
    qfi_EHSI::CDI cdi) [inline]
```

8.39.2.6 setDistance()

```
void WidgetEHSI::setDistance (
    double distance,
    bool visible = false) [inline]
```

8.39.2.7 setHeading()

```
void WidgetEHSI::setHeading (
    double heading) [inline]
```

8.39.2.8 setHeadingSel()

```
void WidgetEHSI::setHeadingSel (
    double headingBug) [inline]
```

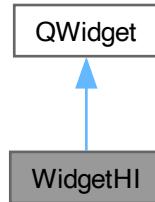
The documentation for this class was generated from the following files:

- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/WidgetEHSI.h
- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/WidgetEHSI.cpp

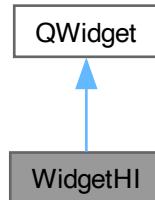
8.40 WidgetHI Class Reference

```
#include <WidgetHI.h>
```

Inheritance diagram for WidgetHI:



Collaboration diagram for WidgetHI:



Public Member Functions

- [WidgetHI \(QWidget *parent=Q_NULLPTR\)](#)
- [~WidgetHI \(\)](#)
- [void redraw \(\)](#)
- [void reinit \(\)](#)
- [void setHeading \(double heading\)](#)

8.40.1 Constructor & Destructor Documentation

8.40.1.1 WidgetHI()

```
WidgetHI::WidgetHI (
    QWidget * parent = Q_NULLPTR) [explicit]
```

Here is the call graph for this function:



Here is the caller graph for this function:



8.40.1.2 ~WidgetHI()

```
WidgetHI::~WidgetHI ()
```

8.40.2 Member Function Documentation

8.40.2.1 redraw()

```
void WidgetHI::redraw () [inline]
```

8.40.2.2 reinit()

```
void WidgetHI::reinit () [inline]
```

8.40.2.3 setHeading()

```
void WidgetHI::setHeading (
    double heading) [inline]
```

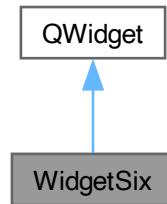
The documentation for this class was generated from the following files:

- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/WidgetHI.h
- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/WidgetHI.cpp

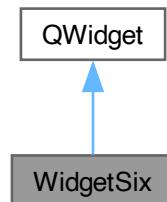
8.41 WidgetSix Class Reference

```
#include <WidgetSix.h>
```

Inheritance diagram for WidgetSix:



Collaboration diagram for WidgetSix:



Public Member Functions

- [WidgetSix \(QWidget *parent=Q_NULLPTR\)](#)
- [~WidgetSix \(\)](#)
- [WidgetAI * getAI \(\)](#)
- [WidgetALT * getALT \(\)](#)
- [WidgetASI * getASI \(\)](#)
- [WidgetHI * getHI \(\)](#)
- [WidgetTC * getTC \(\)](#)
- [WidgetVSI * getVSI \(\)](#)
- [WidgetEADI * getEADI \(\)](#)

8.41.1 Constructor & Destructor Documentation

8.41.1.1 WidgetSix()

```
WidgetSix::WidgetSix (
    QWidget * parent = Q_NULLPTR) [explicit]
```

Here is the call graph for this function:



Here is the caller graph for this function:



8.41.1.2 ~WidgetSix()

```
WidgetSix::~WidgetSix ()
```

8.41.2 Member Function Documentation

8.41.2.1 getAI()

```
WidgetAI * WidgetSix::getAI () [inline]
```

8.41.2.2 getALT()

```
WidgetALT * WidgetSix::getALT () [inline]
```

8.41.2.3 getASI()

```
WidgetASI * WidgetSix::getASI () [inline]
```

8.41.2.4 getEADI()

```
WidgetEADI * WidgetSix::getEADI () [inline]
```

8.41.2.5 getHI()

```
WidgetHI * WidgetSix::getHI () [inline]
```

8.41.2.6 getTC()

```
WidgetTC * WidgetSix::getTC () [inline]
```

8.41.2.7 getVSI()

```
WidgetVSI * WidgetSix::getVSI () [inline]
```

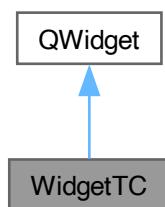
The documentation for this class was generated from the following files:

- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/WidgetSix.h
- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/WidgetSix.cpp

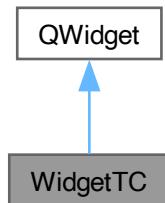
8.42 WidgetTC Class Reference

```
#include <WidgetTC.h>
```

Inheritance diagram for WidgetTC:



Collaboration diagram for WidgetTC:



Public Member Functions

- `WidgetTC (QWidget *parent=Q_NULLPTR)`
- `~WidgetTC ()`
- `void redraw ()`
- `void reinit ()`
- `void setTurnRate (double turnRate)`
- `void setSlipSkid (double slipSkid)`

8.42.1 Constructor & Destructor Documentation

8.42.1.1 WidgetTC()

```
WidgetTC::WidgetTC (
    QWidget * parent = Q_NULLPTR) [explicit]
```

Here is the call graph for this function:



Here is the caller graph for this function:



8.42.1.2 ~WidgetTC()

```
WidgetTC::~WidgetTC ()
```

8.42.2 Member Function Documentation

8.42.2.1 redraw()

```
void WidgetTC::redraw () [inline]
```

8.42.2.2 reinit()

```
void WidgetTC::reinit () [inline]
```

8.42.2.3 setSlipSkid()

```
void WidgetTC::setSlipSkid (
    double slipSkid) [inline]
```

8.42.2.4 setTurnRate()

```
void WidgetTC::setTurnRate (
    double turnRate) [inline]
```

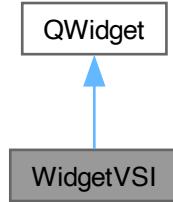
The documentation for this class was generated from the following files:

- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/WidgetTC.h
- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/WidgetTC.cpp

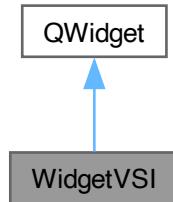
8.43 WidgetVSI Class Reference

```
#include <WidgetVSI.h>
```

Inheritance diagram for WidgetVSI:



Collaboration diagram for WidgetVSI:



Public Member Functions

- [WidgetVSI \(QWidget *parent=Q_NULLPTR\)](#)
- [~WidgetVSI \(\)](#)
- [void redraw \(\)](#)
- [void reinit \(\)](#)
- [void setClimbRate \(double climbRate\)](#)

8.43.1 Constructor & Destructor Documentation

8.43.1.1 WidgetVSI()

```
WidgetVSI::WidgetVSI (
    QWidget * parent = Q_NULLPTR) [explicit]
```

Here is the call graph for this function:



Here is the caller graph for this function:



8.43.1.2 ~WidgetVSI()

```
WidgetVSI::~WidgetVSI ()
```

8.43.2 Member Function Documentation

8.43.2.1 redraw()

```
void WidgetVSI::redraw () [inline]
```

8.43.2.2 reinit()

```
void WidgetVSI::reinit () [inline]
```

8.43.2.3 setClimbRate()

```
void WidgetVSI::setClimbRate (
    double climbRate) [inline]
```

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

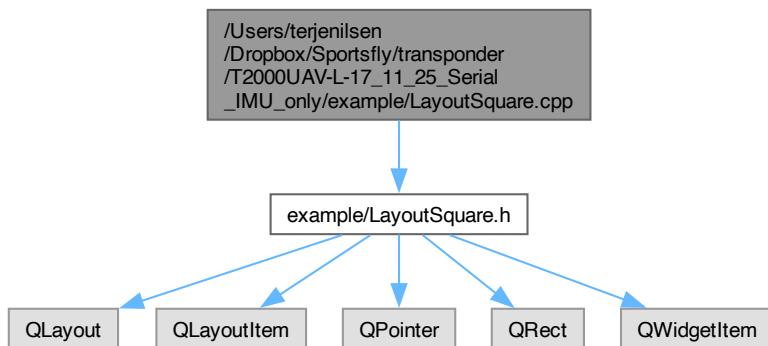
- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/WidgetVSI.h
- /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/WidgetVSI.cpp

Chapter 9

File Documentation

9.1 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/LayoutSquare.cpp File Reference

```
#include <example/LayoutSquare.h>
Include dependency graph for LayoutSquare.cpp:
```



Functions

- int [margin](#) (void)

9.1.1 Function Documentation

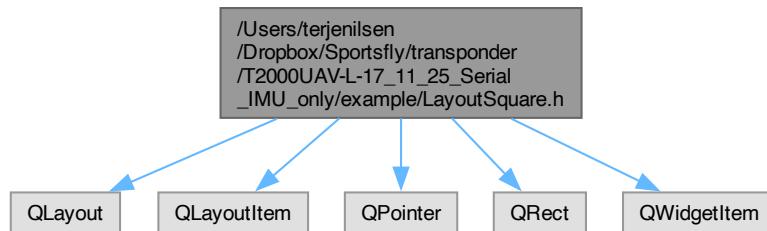
9.1.1.1 margin()

```
int margin (
    void )
```

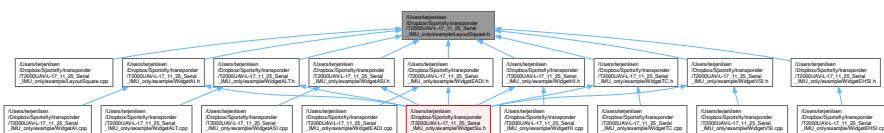
9.2 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_←
11_25_Serial_IMU_only/example/LayoutSquare.h File Reference

```
#include <QLayout>
#include <QLayoutItem>
#include <QPointer>
#include < QRect>
#include <QWidgetItem>
Include dependency graph for LayoutSquare.h:
```

Include dependency graph for LayoutSquare.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- class [LayoutSquare](#)
This class provides constant aspect ratio (square) layout for a single widget.

9.3 LayoutSquare.h

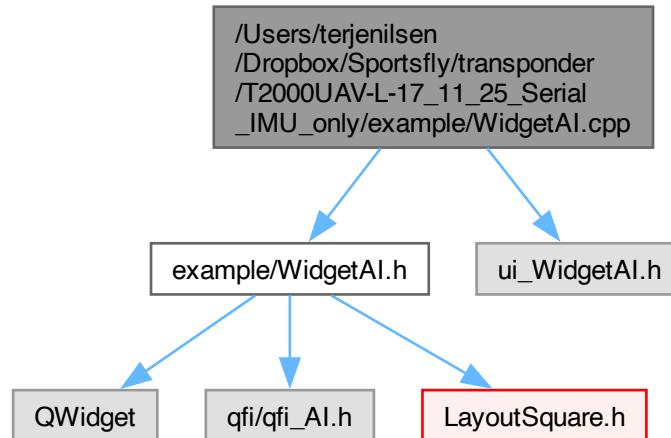
[Go to the documentation of this file.](#)

```
00001 /*****  
00002 * Copyright (C) 2021 Marek M. Cel  
00003 *  
00004 * Permission is hereby granted, free of charge, to any person obtaining  
00005 * a copy of this software and associated documentation files (the "Software"),  
00006 * to deal in the Software without restriction, including without limitation  
00007 * the rights to use, copy, modify, merge, publish, distribute, sublicense,  
00008 * and/or sell copies of the Software, and to permit persons to whom  
00009 * the Software is furnished to do so, subject to the following conditions:  
00010 *  
00011 * The above copyright notice and this permission notice shall be included  
00012 * in all copies or substantial portions of the Software.  
00013 *  
00014 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS  
00015 * OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
```

```
00016 * FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL
00017 * THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
00018 * LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
00019 * OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS
00020 * IN THE SOFTWARE.
00021 ****
00022 #ifndef LAYOUTSQUARE_H
00023 #define LAYOUTSQUARE_H
00024
00025
00026
00027 #include <QLayout>
00028 #include <QLayoutItem>
00029 #include <QPointer>
00030 #include <QRect>
00031 #include <QWidgetItem>
00032
00033
00034
00035 class LayoutSquare : public QLayout
00036 {
00037     Q_OBJECT
00038
00039 public:
00040     explicit LayoutSquare( QWidget *parent, int spacing = -1 );
00041     explicit LayoutSquare( int spacing = -1 );
00042
00043     virtual ~LayoutSquare();
00044
00045     void addItem( QLayoutItem *item );
00046
00047     void addWidget( QWidget *widget );
00048
00049     int count() const;
00050
00051     Qt::Orientations expandingDirections() const;
00052
00053     QRect geometry2();
00054
00055     bool hasHeightForWidth() const;
00056
00057     bool hasItem() const;
00058
00059     QLayoutItem* itemAt( int index ) const;
00060
00061     QSize minimumSize() const;
00062
00063     QLayoutItem* replaceItem( QLayoutItem *item );
00064
00065     void setGeometry( const QRect &rect );
00066
00067     QSize sizeHint() const;
00068
00069     QLayoutItem * take();
00070
00071     QLayoutItem * takeAt( int index );
00072
00073 private:
00074     QLayoutItem *_item;
00075
00076     QRect *_rectLast;
00077     QRect *_geometry;
00078
00079     bool areRectsEqual( const QRect &rect_1, const QRect &rect_2 ) const;
00080
00081     QPoint calculateCenterPnt( QSize fromSize, QSize itemSize ) const;
00082
00083     QSize calculateProperSize( QSize fromSize ) const;
00084
00085     void init( int spacing );
00086
00087     void setRectLast( const QRect &rect );
00088
00089 };
00090
00091
00092
00093
00094
00095
00096
00097
00098
00099
00100
00101
00102
00103
00104
00105
00106
00107
00108
00109
00110
00111
00112 };
00113
00114
00115
00116 #endif // LAYOUTSQUARE_H
```

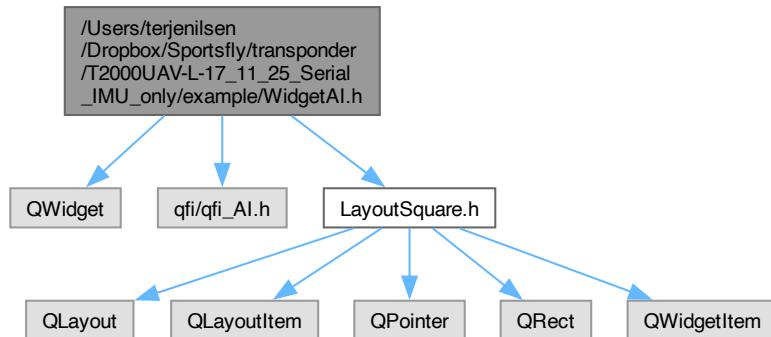
9.4 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/WidgetAI.cpp File Reference

```
#include <example/WidgetAI.h>
#include <ui_WidgetAI.h>
Include dependency graph for WidgetAI.cpp:
```

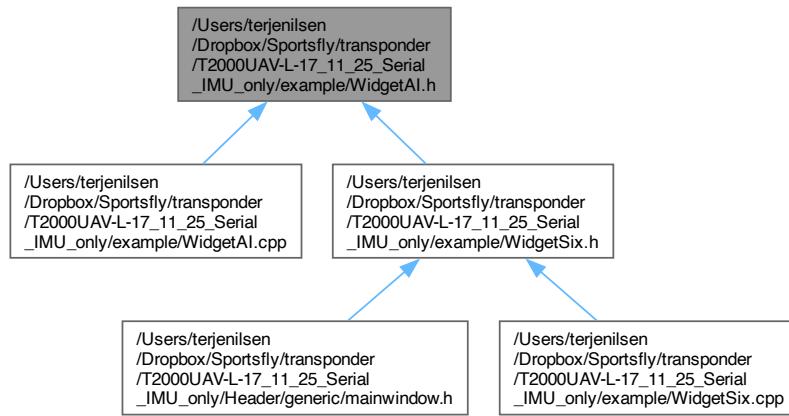


9.5 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/WidgetAI.h File Reference

```
#include <QWidget>
#include <qfi/qfi_AI.h>
#include "LayoutSquare.h"
Include dependency graph for WidgetAI.h:
```



This graph shows which files directly or indirectly include this file:



Data Structures

- class [WidgetAI](#)

Namespaces

- namespace [Ui](#)

9.6 WidgetAI.h

[Go to the documentation of this file.](#)

```

00001 /*****\n00002 * Copyright (C) 2021 Marek M. Cel\n00003 *\n00004 * Permission is hereby granted, free of charge, to any person obtaining\n00005 * a copy of this software and associated documentation files (the "Software"),\n00006 * to deal in the Software without restriction, including without limitation\n00007 * the rights to use, copy, modify, merge, publish, distribute, sublicense,\n00008 * and/or sell copies of the Software, and to permit persons to whom\n00009 * the Software is furnished to do so, subject to the following conditions:\n00010 *\n00011 * The above copyright notice and this permission notice shall be included\n00012 * in all copies or substantial portions of the Software.\n00013 *\n00014 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS\n00015 * OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,\n00016 * FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL\n00017 * THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER\n00018 * LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,\n00019 * OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS\n00020 * IN THE SOFTWARE.\n00021 *****\n00022 #ifndef WIDGETAI_H\n00023 #define WIDGETAI_H\n00024\n00025\n00026 #include <QWidget>\n00027 #include <qfi/qfi_AI.h>\n00028\n00029 #include "LayoutSquare.h"\n00030\n00031\n00032
  
```

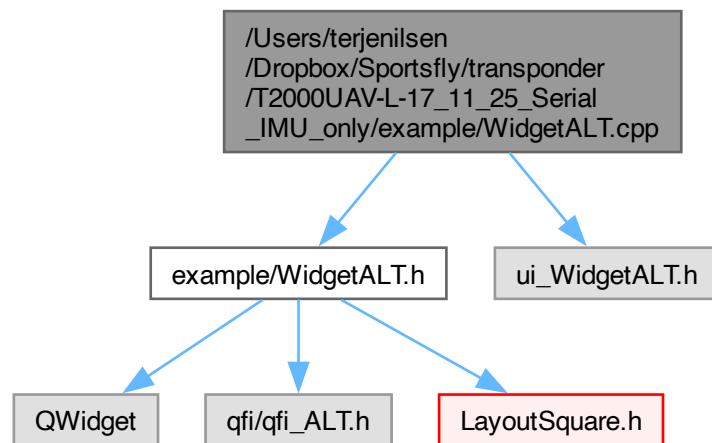
```

00034 QT_BEGIN_NAMESPACE
00035 //namespace Ui { class SCREEN; }
00036 namespace Ui{class WidgetAI;}
00037 QT_END_NAMESPACE
00038
00040
00041 class WidgetAI : public QWidget
00042 {
00043     Q_OBJECT
00044
00045 public:
00046     explicit WidgetAI( QWidget *parent = Q_NULLPTR );
00047     ~WidgetAI();
00048
00049     inline void redraw() { _ai->redraw(); }
00050     inline void reinit() { _ai->reinit(); }
00051
00052     inline void setRoll( double roll )
00053     {
00054         _ai->setRoll( roll );
00055     }
00056
00057     inline void setPitch( double pitch )
00058     {
00059         _ai->setPitch( pitch );
00060     }
00061
00062 }
00063
00064 private:
00065
00066     Ui::WidgetAI *_ui;
00067     qfi_AI      *_ai;
00068     LayoutSquare *_layoutSq;
00069
00070     void setupUi();
00071 };
00072
00073
00074 #endif // WIDGETAI_H

```

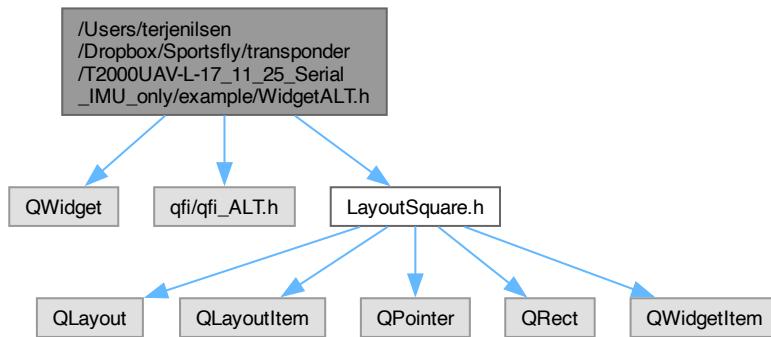
9.7 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_← 11_25_Serial_IMU_only/example/WidgetALT.cpp File Reference

#include <example/WidgetALT.h>
#include <ui_WidgetALT.h>
Include dependency graph for WidgetALT.cpp:

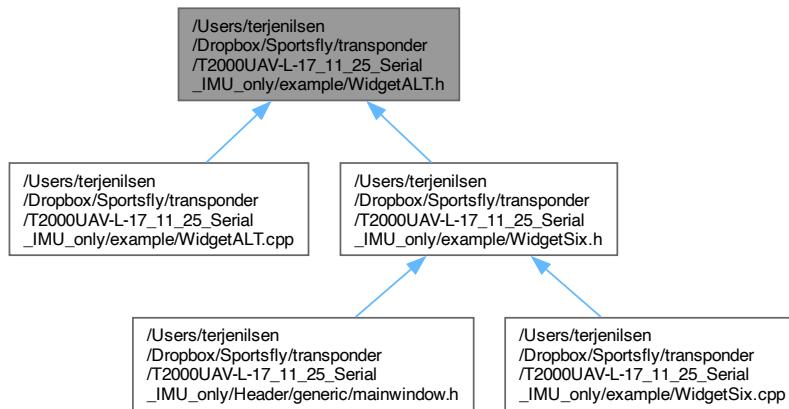


9.8 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/WidgetALT.h File Reference ← 11_25_Serial_IMU_only/example/WidgetALT.h File Reference

```
#include <QWidget>
#include <qfi/qfi_ALT.h>
#include "LayoutSquare.h"
Include dependency graph for WidgetALT.h:
```



This graph shows which files directly or indirectly include this file:



Data Structures

- class [WidgetALT](#)

Namespaces

- namespace [Ui](#)

9.9 WidgetALT.h

[Go to the documentation of this file.](#)

```

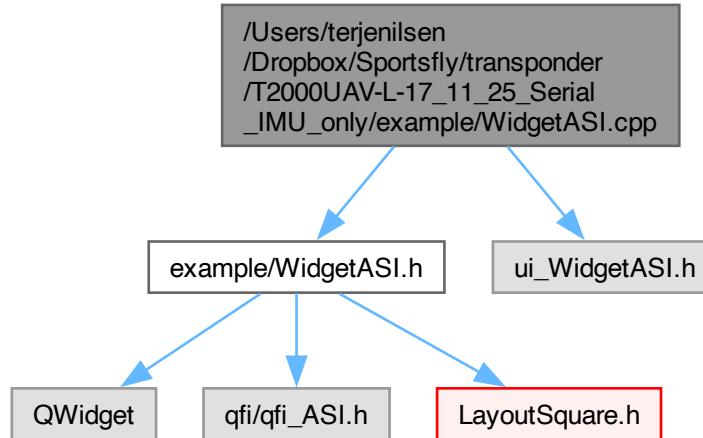
00001 /*****  
00002 * Copyright (C) 2021 Marek M. Cel  
00003 *  
00004 * Permission is hereby granted, free of charge, to any person obtaining  
00005 * a copy of this software and associated documentation files (the "Software"),  
00006 * to deal in the Software without restriction, including without limitation  
00007 * the rights to use, copy, modify, merge, publish, distribute, sublicense,  
00008 * and/or sell copies of the Software, and to permit persons to whom  
00009 * the Software is furnished to do so, subject to the following conditions:  
00010 *  
00011 * The above copyright notice and this permission notice shall be included  
00012 * in all copies or substantial portions of the Software.  
00013 *  
00014 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS  
00015 * OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,  
00016 * FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL  
00017 * THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER  
00018 * LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,  
00019 * OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS  
00020 * IN THE SOFTWARE.  
00021 *****/  
00022 #ifndef WIDGETALT_H  
00023 #define WIDGETALT_H  
00024  
00026  
00027 #include <QWidget>  
00028  
00029 #include <qfi/qfi_ALT.h>  
00030  
00031 #include "LayoutSquare.h"  
00032  
00034  
00035 namespace Ui  
00036 {  
00037     class WidgetALT;  
00038 }  
00039  
00041  
00042 class WidgetALT : public QWidget  
00043 {  
00044     Q_OBJECT  
00045  
00046 public:  
00047     explicit WidgetALT( QWidget *parent = Q_NULLPTR );  
00049  
00050     ~WidgetALT();  
00051  
00052     inline void redraw() { _alt->redraw(); }  
00053     inline void reinit() { _alt->reinit(); }  
00054  
00055     inline void setAltitude( double altitude )  
00056     {  
00057         _alt->setAltitude( altitude );  
00058     }  
00059  
00060     inline void setPressure( double pressure )  
00061     {  
00062         _alt->setPressure( pressure );  
00063     }  
00064  
00065 private:  
00066  
00067     Ui::WidgetALT *_ui;  
00068     qfi_ALT      *_alt;  
00069     LayoutSquare *_layoutSq;  
00070  
00071     void setupUi();  
00072 };  
00073  
00075  
00076 #endif // WIDGETALT_H

```

9.10 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/WidgetASI.cpp File Reference ←

11_25_Serial_IMU_only/example/WidgetASI.cpp File Reference

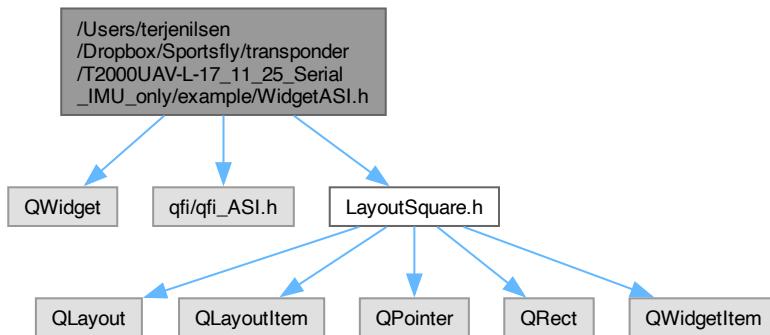
```
#include <example/WidgetASI.h>
#include <ui_WidgetASI.h>
Include dependency graph for WidgetASI.cpp:
```



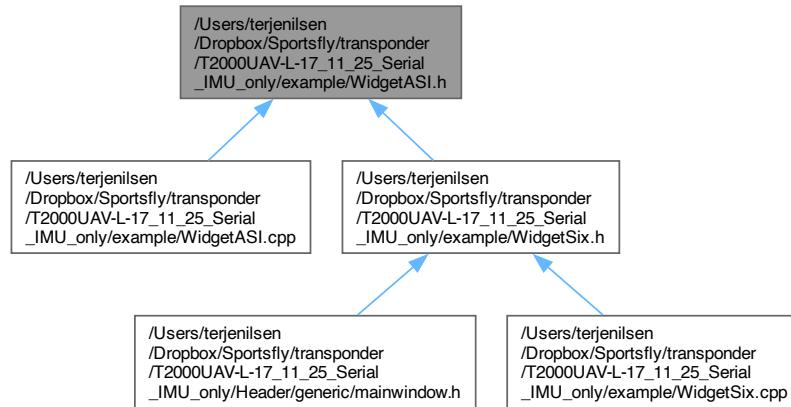
9.11 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/WidgetASI.h File Reference ←

11_25_Serial_IMU_only/example/WidgetASI.h File Reference

```
#include <QWidget>
#include <qfi/qfi_AS1.h>
#include "LayoutSquare.h"
Include dependency graph for WidgetASI.h:
```



This graph shows which files directly or indirectly include this file:



Data Structures

- class [WidgetASI](#)

Namespaces

- namespace [Ui](#)

9.12 WidgetASI.h

[Go to the documentation of this file.](#)

```

00001 /*****\n00002 * Copyright (C) 2021 Marek M. Cel\n00003 *\n00004 * Permission is hereby granted, free of charge, to any person obtaining\n00005 * a copy of this software and associated documentation files (the "Software"),\n00006 * to deal in the Software without restriction, including without limitation\n00007 * the rights to use, copy, modify, merge, publish, distribute, sublicense,\n00008 * and/or sell copies of the Software, and to permit persons to whom\n00009 * the Software is furnished to do so, subject to the following conditions:\n00010 *\n00011 * The above copyright notice and this permission notice shall be included\n00012 * in all copies or substantial portions of the Software.\n00013 *\n00014 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS\n00015 * OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,\n00016 * FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL\n00017 * THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER\n00018 * LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,\n00019 * OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS\n00020 * IN THE SOFTWARE.\n00021 *****\n00022 #ifndef WIDGETASI_H\n00023 #define WIDGETASI_H\n00024\n00026\n00027 #include <QWidget>\n00028\n00029 #include <qfi/qfi_ASI.h>\n00030\n00031 #include "LayoutSquare.h"\n00032\n00034
  
```

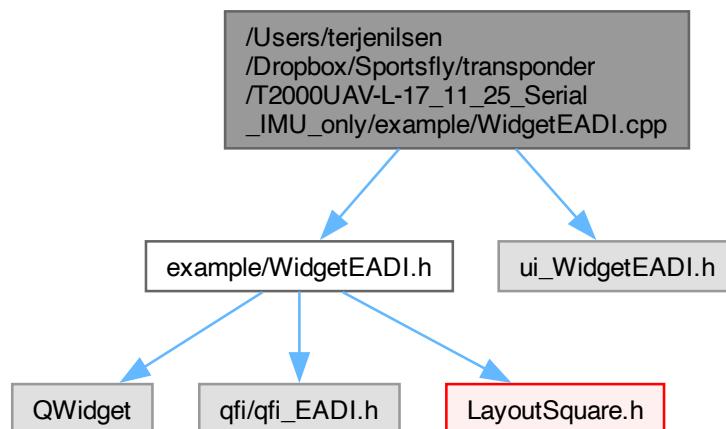
```

00035 namespace Ui
00036 {
00037     class WidgetASI;
00038 }
00039
00041
00042 class WidgetASI : public QWidget
00043 {
00044     Q_OBJECT
00045
00046 public:
00047
00048     explicit WidgetASI( QWidget *parent = Q_NULLPTR );
00049
00050     ~WidgetASI();
00051
00052     inline void redraw() { _asi->redraw(); }
00053     inline void reinit() { _asi->reinit(); }
00054
00055     inline void setAirspeed( double airspeed )
00056     {
00057         _asi->setAirspeed( airspeed );
00058     }
00059
00060 private:
00061
00062     Ui::WidgetASI *_ui;
00063     qfi_ASI      *_asi;
00064     LayoutSquare *_layoutSq;
00065
00066     void setupUi();
00067 };
00068
00069
00070
00071 #endif // WIDGETASI_H

```

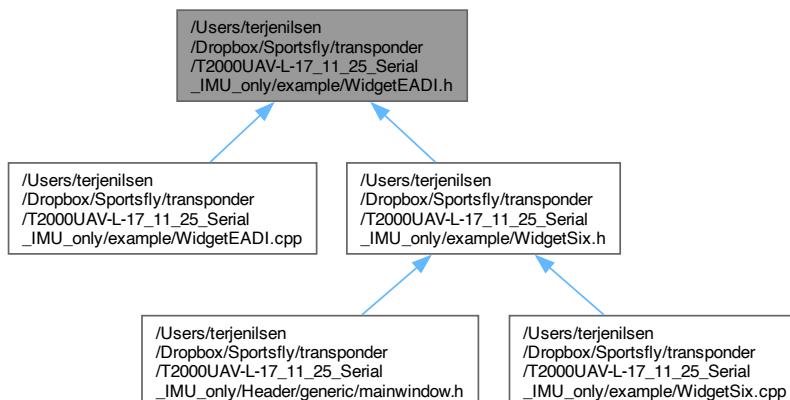
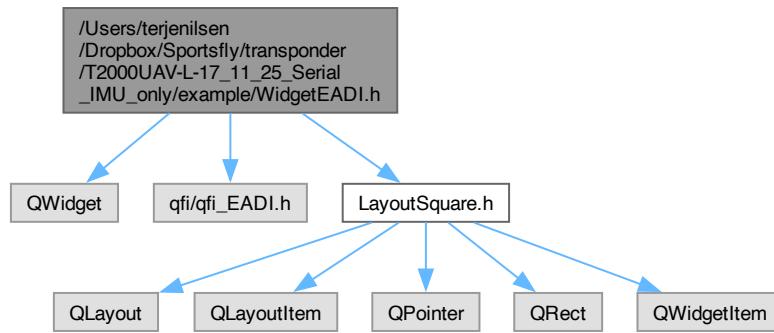
9.13 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/WidgetEADI.cpp File Reference ←

```
#include <example/WidgetEADI.h>
#include <ui_WidgetEADI.h>
Include dependency graph for WidgetEADI.cpp:
```



9.14 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/WidgetEADI.h File Reference

```
#include <QWidget>
#include <qfi/qfi_EADI.h>
#include "LayoutSquare.h"
Include dependency graph for WidgetEADI.h:
```



Data Structures

- class [WidgetEADI](#)

Namespaces

- namespace [Ui](#)

9.15 WidgetEADI.h

[Go to the documentation of this file.](#)

```
00001 /*****  
00002 * Copyright (C) 2021 Marek M. Cel  
00003 *  
00004 * Permission is hereby granted, free of charge, to any person obtaining  
00005 * a copy of this software and associated documentation files (the "Software"),  
00006 * to deal in the Software without restriction, including without limitation  
00007 * the rights to use, copy, modify, merge, publish, distribute, sublicense,  
00008 * and/or sell copies of the Software, and to permit persons to whom  
00009 * the Software is furnished to do so, subject to the following conditions:  
00010 *  
00011 * The above copyright notice and this permission notice shall be included  
00012 * in all copies or substantial portions of the Software.  
00013 *  
00014 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS  
00015 * OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,  
00016 * FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL  
00017 * THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER  
00018 * LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,  
00019 * OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS  
00020 * IN THE SOFTWARE.  
00021 *****/  
00022 #ifndef WIDGETEADI_H  
00023 #define WIDGETEADI_H  
00024  
00026  
00027 #include <QWidget>  
00028  
00029 #include <qfi/qfi_EADI.h>  
00030  
00031 #include "LayoutSquare.h"  
00032  
00033  
00034  
00035 namespace Ui  
00036 {  
00037     class WidgetEADI;  
00038 }  
00039  
00040  
00041 class WidgetEADI : public QWidget  
00042 {  
00043     Q_OBJECT  
00044  
00045     public:  
00046         explicit WidgetEADI( QWidget *parent = Q_NULLPTR );  
00047  
00048         virtual ~WidgetEADI();  
00049  
00050         inline void reinit() { _eadi->reinit(); }  
00051  
00052         inline void redraw() { _eadi->redraw(); }  
00053  
00054         inline void setFltMode( qfi_EADI::FltMode fltMode )  
00055         {  
00056             _eadi->setFltMode( fltMode );  
00057         }  
00058  
00059         inline void setSpdMode( qfi_EADI::SpdMode spdMode )  
00060         {  
00061             _eadi->setSpdMode( spdMode );  
00062         }  
00063  
00064         inline void setLNAV( qfi_EADI::LNAV lnav )  
00065         {  
00066             _eadi->setLNAV( lnav );  
00067         }  
00068  
00069         inline void setVNAV( qfi_EADI::VNAV vnav )  
00070         {  
00071             _eadi->setVNAV( vnav );  
00072         }  
00073  
00074         inline void setRoll( double roll )  
00075         {  
00076             _eadi->setRoll( roll );  
00077         }  
00078  
00079         inline void setPitch( double pitch )  
00080         {  
00081             _eadi->setPitch( pitch );  
00082         }  
00083  
00084  
00085         inline void setBank( double bank )  
00086         {  
00087             _eadi->setBank( bank );  
00088         }  
00089 }
```

```

00090     inline void setFPM( double aoa, double sideslip )
00091     {
00092         _eadi->setFPM( aoa, sideslip );
00093     }
00094
00095     inline void setSlipSkid( double slipSkid )
00096     {
00097         _eadi->setSlipSkid( slipSkid );
00098     }
00099
00103     inline void setTurnRate( double turnRate )
00104     {
00105         _eadi->setTurnRate( turnRate );
00106     }
00107
00113     inline void setDots( double dotH, double dotV, bool visibleH, bool visibleV )
00114     {
00115         _eadi->setDots( dotH, dotV, visibleH, visibleV );
00116     }
00117
00122     inline void setFD( double roll, double pitch, bool visible = true )
00123     {
00124         _eadi->setFD( roll, pitch, visible );
00125     }
00126
00128     inline void setStall( bool stall )
00129     {
00130         _eadi->setStall( stall );
00131     }
00132
00133     inline void setAltitude( double altitude )
00134     {
00135         _eadi->setAltitude( altitude );
00136     }
00137
00138     inline void setPressure( double pressure, qfi_EADI::PressureMode pressMode )
00139     {
00140         _eadi->setPressure( pressure, pressMode );
00141     }
00142
00143     inline void setAirspeed( double airspeed )
00144     {
00145         _eadi->setAirspeed( airspeed );
00146     }
00147
00148     inline void setMachNo( double machNo )
00149     {
00150         _eadi->setMachNo( machNo );
00151     }
00152
00153     inline void setHeading( double heading )
00154     {
00155         _eadi->setHeading( heading );
00156     }
00157
00158     inline void setClimbRate( double climbRate )
00159     {
00160         _eadi->setClimbRate( climbRate );
00161     }
00162
00164     inline void setAirspeedSel( double airspeed )
00165     {
00166         _eadi->setAirspeedSel( airspeed );
00167     }
00168
00170     inline void setAltitudeSel( double altitude )
00171     {
00172         _eadi->setAltitudeSel( altitude );
00173     }
00174
00176     inline void setHeadingSel( double heading )
00177     {
00178         _eadi->setHeadingSel( heading );
00179     }
00180
00181 //    inline void setIdentifier( const QString &ident, bool isVisible )
00182 //    {{
00183 //        _eadi->setIdent( ident, isVisible );
00184 //    }
00185
00186 //    inline void setDistance( double dist, bool isVisible )
00187 //    {{
00188 //        _eadi->setDistance( dist, isVisible );
00189 //    }
00190
00191     inline void setVfe( double vfe )
00192     {

```

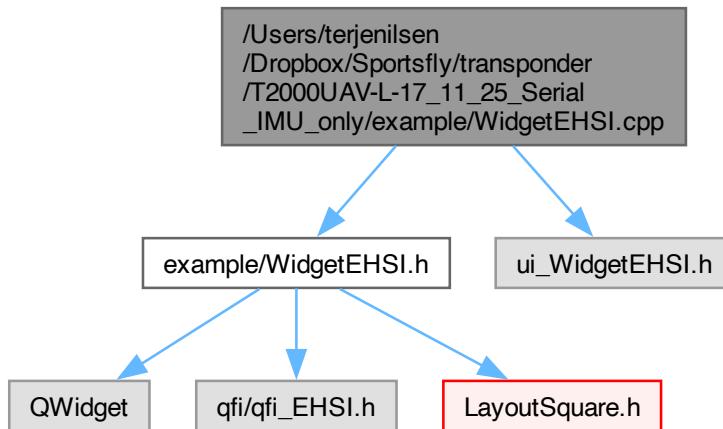
```

00193     _eadi->setVfe( vfe );
00194 }
00195
00196 inline void setVne( double vne )
00197 {
00198     _eadi->setVne( vne );
00199 }
00200
00201 private:
00202
00203     Ui::WidgetEADI *_ui;
00204     qfi_EADI      *_eadi;
00205     LayoutSquare   *_layoutsSq;
00206
00207     void setupUi();
00208 };
00209
00211
00212 #endif // WIDGETEADI_H

```

9.16 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/WidgetEHSI.cpp File Reference

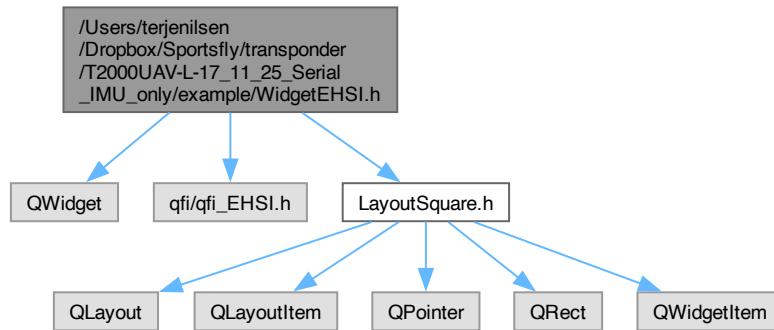
```
#include <example/WidgetEHSI.h>
#include <ui_WidgetEHSI.h>
Include dependency graph for WidgetEHSI.cpp:
```



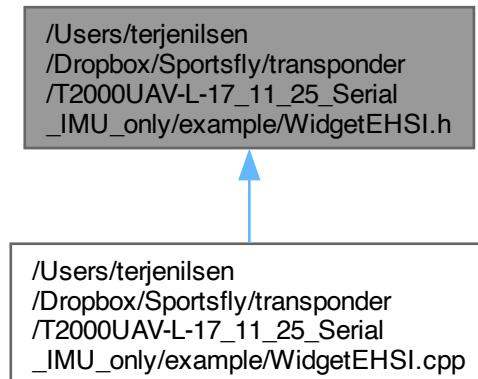
9.17 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/WidgetEHSI.h File Reference

```
#include <QWidget>
#include <qfi/qfi_EHSI.h>
```

```
#include "LayoutSquare.h"
Include dependency graph for WidgetEHSI.h:
```



This graph shows which files directly or indirectly include this file:



Data Structures

- class [WidgetEHSI](#)

Namespaces

- namespace [Ui](#)

9.18 WidgetEHSI.h

Go to the documentation of this file.

```
00001 /*****  
00002 * Copyright (C) 2021 Marek M. Cel  
00003 *  
00004 * Permission is hereby granted, free of charge, to any person obtaining  
00005 * a copy of this software and associated documentation files (the "Software"),  
00006 * to deal in the Software without restriction, including without limitation  
00007 * the rights to use, copy, modify, merge, publish, distribute, sublicense,  
00008 * and/or sell copies of the Software, and to permit persons to whom  
00009 * the Software is furnished to do so, subject to the following conditions:  
00010 *  
00011 * The above copyright notice and this permission notice shall be included  
00012 * in all copies or substantial portions of the Software.  
00013 *  
00014 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS  
00015 * OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,  
00016 * FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL  
00017 * THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER  
00018 * LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,  
00019 * OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS  
00020 * IN THE SOFTWARE.  
00021 ****/  
00022 #ifndef WIDGETEHSI_H  
00023 #define WIDGETEHSI_H  
00024  
00026  
00027 #include <QWidget>  
00028  
00029 #include <qfi/qfi_EHSI.h>  
00030  
00031 #include "LayoutSquare.h"  
00032  
00033  
00034  
00035 namespace Ui  
00036 {  
00037     class WidgetEHSI;  
00038 }  
00039  
00040  
00041 class WidgetEHSI : public QWidget  
00042 {  
00043     Q_OBJECT  
00044  
00045     public:  
00046         explicit WidgetEHSI( QWidget *parent = Q_NULLPTR );  
00047  
00048         virtual ~WidgetEHSI();  
00049  
00050         inline void reinit() { _ehsi->reinit(); }  
00051  
00052         inline void redraw() { _ehsi->redraw(); }  
00053  
00054         inline void setHeading( double heading )  
00055         {  
00056             _ehsi->setHeading( heading );  
00057         }  
00058  
00059         inline void setCourse( double course )  
00060         {  
00061             _ehsi->setCourse( course );  
00062         }  
00063  
00064         inline void setBearing( double bearing, bool visible = false )  
00065         {  
00066             _ehsi->setBearing( bearing, visible );  
00067         }  
00068  
00069         inline void setDeviation( double deviation, qfi_EHSI::CDI cdi )  
00070         {  
00071             _ehsi->setDeviation( deviation, cdi );  
00072         }  
00073  
00074         inline void setDistance( double distance, bool visible = false )  
00075         {  
00076             _ehsi->setDistance( distance, visible );  
00077         }  
00078  
00079         inline void setHeadingSel( double headingBug )  
00080         {  
00081             _ehsi->setHeadingSel( headingBug );  
00082         }  
00083  
00084         inline void setHeadingSel( headingBug );  
00085 }
```

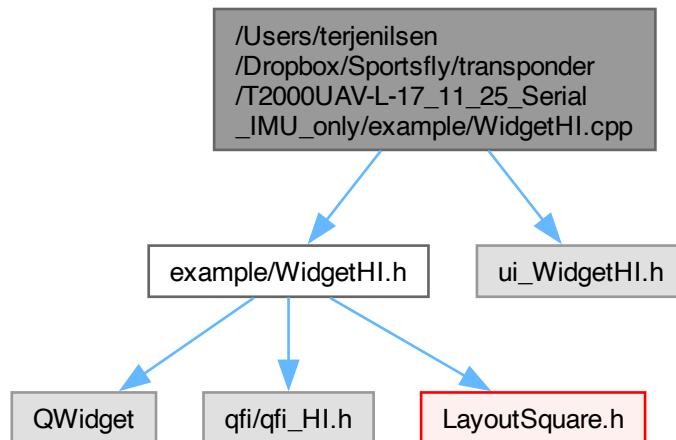
```

00086 private:
00087
00088     Ui::WidgetEHSI *_ui;
00089     qfi_EHSI      *_ehsi;
00090     LayoutSquare   *_layoutSq;
00091
00092     void setupUi();
00093 };
00094
00095
00096 #endif // WIDGETEHSI_H

```

9.19 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/WidgetHI.cpp File Reference

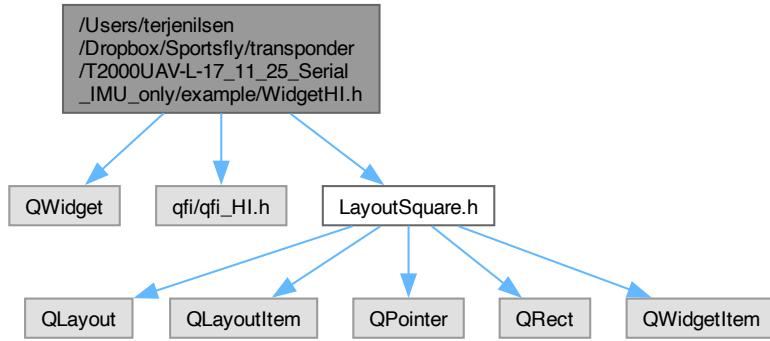
```
#include <example/WidgetHI.h>
#include <ui_WidgetHI.h>
Include dependency graph for WidgetHI.cpp:
```



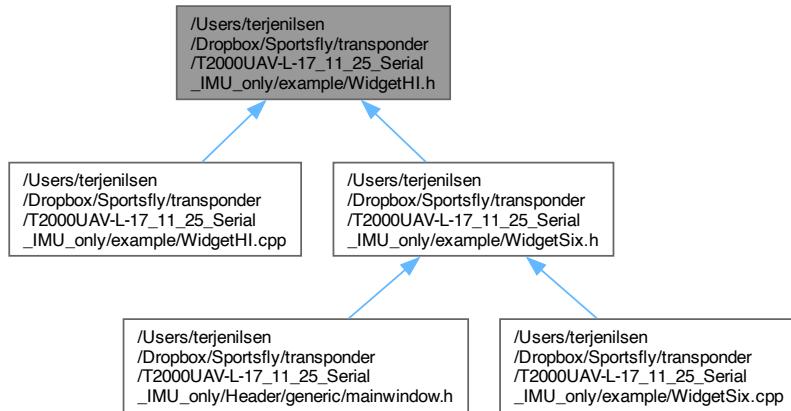
9.20 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/WidgetHI.h File Reference

```
#include <QWidget>
#include <qfi/qfi_HI.h>
#include "LayoutSquare.h"
```

Include dependency graph for WidgetHI.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- class [WidgetHI](#)

Namespaces

- namespace [Ui](#)

9.21 WidgetHI.h

[Go to the documentation of this file.](#)

```

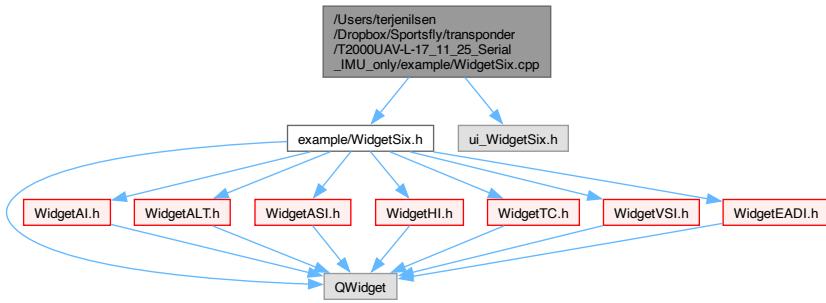
00001 /*****  
00002 * Copyright (C) 2021 Marek M. Cel  
00003 *  
00004 * Permission is hereby granted, free of charge, to any person obtaining  
00005 * a copy of this software and associated documentation files (the "Software"),  
00006 * to deal in the Software without restriction, including without limitation  
00007 * the rights to use, copy, modify, merge, publish, distribute, sublicense,  
00008 * and/or sell copies of the Software, and to permit persons to whom  
00009 * the Software is furnished to do so, subject to the following conditions:  
00010 *  
00011 * The above copyright notice and this permission notice shall be included  
00012 * in all copies or substantial portions of the Software.  
00013 *  
00014 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS  
00015 * OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,  
00016 * FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL  
00017 * THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER  
00018 * LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,  
00019 * OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS  
00020 * IN THE SOFTWARE.  
00021 ****/  
00022 #ifndef WIDGETHI_H  
00023 #define WIDGETHI_H  
00024  
00026  
00027 #include <QWidget>  
00028  
00029 #include <qfi/qfi_HI.h>  
00030  
00031 #include "LayoutSquare.h"  
00032  
00034  
00035 namespace Ui  
00036 {  
00037     class WidgetHI;  
00038 }  
00039  
00041  
00042 class WidgetHI : public QWidget  
00043 {  
00044     Q_OBJECT  
00045  
00046 public:  
00047     explicit WidgetHI( QWidget *parent = Q_NULLPTR );  
00049  
00050     ~WidgetHI();  
00051  
00052     inline void redraw() { _hi->redraw(); }  
00053     inline void reinit() { _hi->reinit(); }  
00054  
00055     inline void setHeading( double heading )  
00056     {  
00057         _hi->setHeading( heading );  
00058     }  
00059  
00060 private:  
00061  
00062     Ui::WidgetHI *_ui;  
00063     qfi_HI *_hi;  
00064     LayoutSquare *_layoutSq;  
00065  
00066     void setupUi();  
00067 };  
00068  
00070  
00071 #endif // WIDGETHI_H

```

9.22 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_← 11_25_Serial_IMU_only/example/WidgetSix.cpp File Reference

```
#include <example/WidgetSix.h>  
#include <ui_WidgetSix.h>
```

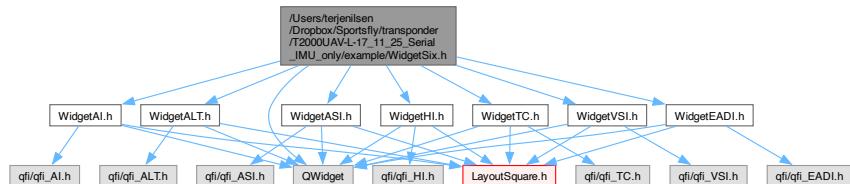
Include dependency graph for WidgetSix.cpp:



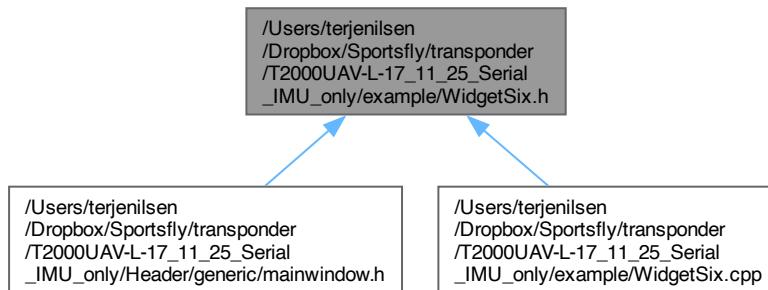
9.23 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/WidgetSix.h File Reference

```
#include <QWidget>
#include "WidgetAI.h"
#include "WidgetALT.h"
#include "WidgetASI.h"
#include "WidgetHI.h"
#include "WidgetTC.h"
#include "WidgetVSI.h"
#include "WidgetEADI.h"
```

Include dependency graph for WidgetSix.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- class [WidgetSix](#)

Namespaces

- namespace [Ui](#)

9.24 WidgetSix.h

[Go to the documentation of this file.](#)

```

00001 /*****  
00002 * Copyright (C) 2021 Marek M. Cel  
00003 *  
00004 * Permission is hereby granted, free of charge, to any person obtaining  
00005 * a copy of this software and associated documentation files (the "Software"),  
00006 * to deal in the Software without restriction, including without limitation  
00007 * the rights to use, copy, modify, merge, publish, distribute, sublicense,  
00008 * and/or sell copies of the Software, and to permit persons to whom  
00009 * the Software is furnished to do so, subject to the following conditions:  
00010 *  
00011 * The above copyright notice and this permission notice shall be included  
00012 * in all copies or substantial portions of the Software.  
00013 *  
00014 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS  
00015 * OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,  
00016 * FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL  
00017 * THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER  
00018 * LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,  
00019 * OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS  
00020 * IN THE SOFTWARE.  
00021 *****  
00022 #ifndef WIDGETSIX_H  
00023 #define WIDGETSIX_H  
00024  
00026  
00027 #include <QWidget>  
00028  
00029 #include "WidgetAI.h"  
00030 #include "WidgetALT.h"  
00031 #include "WidgetASI.h"  
00032 #include "WidgetHI.h"  
00033 #include "WidgetTC.h"  
00034 #include "WidgetVSI.h"  
00035 #include "WidgetEADI.h"  
00036  
00038

```

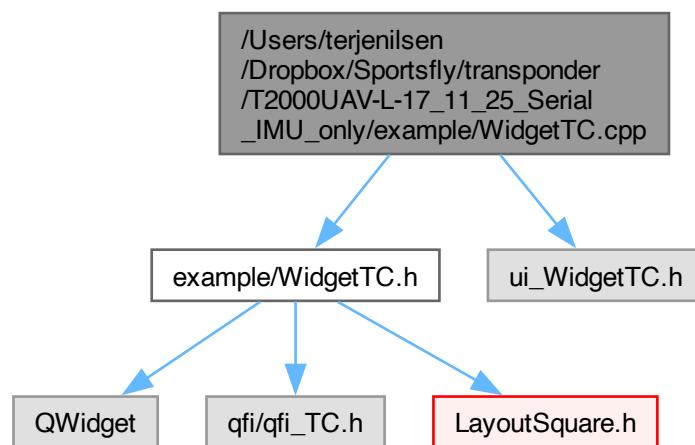
```

00039 namespace Ui
00040 {
00041     class WidgetSix;
00042 }
00043
00045
00046 class WidgetSix : public QWidget
00047 {
00048     Q_OBJECT
00049
00050 public:
00051     explicit WidgetSix( QWidget *parent = Q_NULLPTR );
00052     ~WidgetSix();
00053
00054     WidgetAI * getAI () { return _widgetAI; }
00055     WidgetALT * getALT () { return _widgetALT; }
00056     WidgetASI * getASI () { return _widgetASI; }
00057     WidgetHI * getHI () { return _widgetHI; }
00058     WidgetTC * getTC () { return _widgetTC; }
00059     WidgetVSI * getVSI () { return _widgetVSI; }
00060     WidgetEADI * getEADI () { return _widgetEADI; }
00061
00062 private:
00063     Ui::WidgetSix *_ui;
00064
00065     WidgetAI *_widgetAI;
00066     WidgetALT *_widgetALT;
00067     WidgetASI *_widgetASI;
00068     WidgetHI *_widgetHI;
00069     WidgetTC *_widgetTC;
00070     WidgetVSI *_widgetVSI;
00071     WidgetEADI *_widgetEADI;
00072 };
00073
00074
00075 }
00076
00077
00078
00079 #endif // WIDGETSIX_H

```

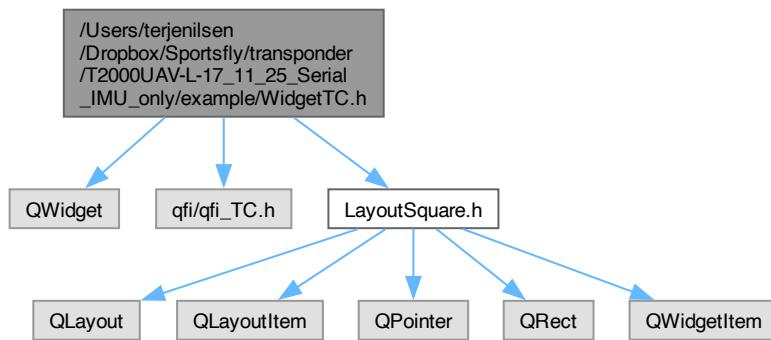
9.25 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/WidgetTC.cpp File Reference

```
#include <example/WidgetTC.h>
#include <ui_WidgetTC.h>
Include dependency graph for WidgetTC.cpp:
```

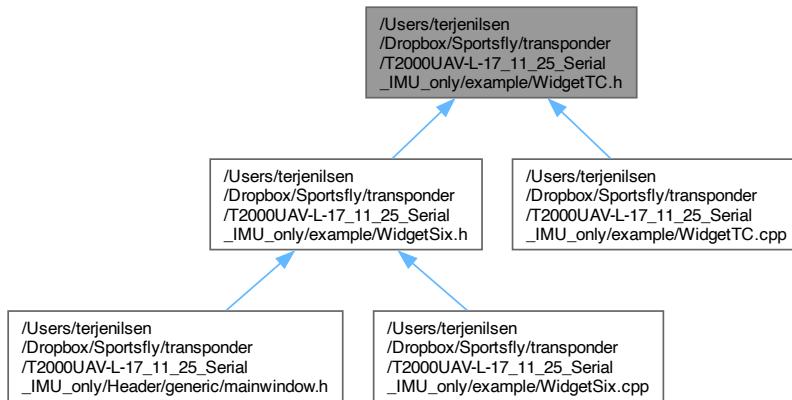


9.26 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/WidgetTC.h File Reference

```
#include <QWidget>
#include <qfi/qfi_TC.h>
#include "LayoutSquare.h"
Include dependency graph for WidgetTC.h:
```



This graph shows which files directly or indirectly include this file:



Data Structures

- class [WidgetTC](#)

Namespaces

- namespace [Ui](#)

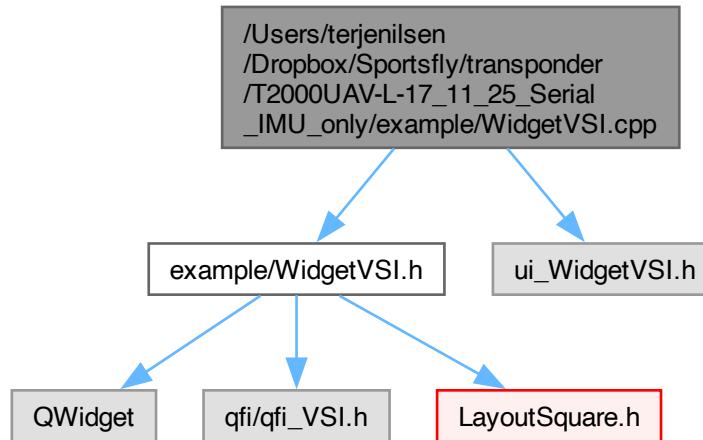
9.27 WidgetTC.h

[Go to the documentation of this file.](#)

```
00001 /*****  
00002 * Copyright (C) 2021 Marek M. Cel  
00003 *  
00004 * Permission is hereby granted, free of charge, to any person obtaining  
00005 * a copy of this software and associated documentation files (the "Software"),  
00006 * to deal in the Software without restriction, including without limitation  
00007 * the rights to use, copy, modify, merge, publish, distribute, sublicense,  
00008 * and/or sell copies of the Software, and to permit persons to whom  
00009 * the Software is furnished to do so, subject to the following conditions:  
00010 *  
00011 * The above copyright notice and this permission notice shall be included  
00012 * in all copies or substantial portions of the Software.  
00013 *  
00014 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS  
00015 * OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,  
00016 * FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL  
00017 * THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER  
00018 * LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,  
00019 * OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS  
00020 * IN THE SOFTWARE.  
00021 ****/  
00022 #ifndef WIDGETTC_H  
00023 #define WIDGETTC_H  
00024  
00026  
00027 #include <QWidget>  
00028  
00029 #include <qfi/qfi_TC.h>  
00030  
00031 #include "LayoutSquare.h"  
00032  
00034  
00035 namespace Ui  
00036 {  
00037     class WidgetTC;  
00038 }  
00039  
00041  
00042 class WidgetTC : public QWidget  
00043 {  
00044     Q_OBJECT  
00045  
00046 public:  
00047     explicit WidgetTC( QWidget *parent = Q_NULLPTR );  
00049  
00050     ~WidgetTC();  
00051  
00052     inline void redraw() { _tc->redraw(); }  
00053     inline void reinit() { _tc->reinit(); }  
00054  
00055     inline void setTurnRate( double turnRate )  
00056     {  
00057         _tc->setTurnRate( turnRate );  
00058     }  
00059  
00060     inline void setSlipSkid( double slipSkid )  
00061     {  
00062         _tc->setSlipSkid( slipSkid );  
00063     }  
00064  
00065 private:  
00066  
00067     Ui::WidgetTC *_ui;  
00068     qfi_TC          *_tc;  
00069     LayoutSquare *_layoutSq;  
00070  
00071     void setupUi();  
00072 };  
00073  
00075  
00076 #endif // WIDGETTC_H
```

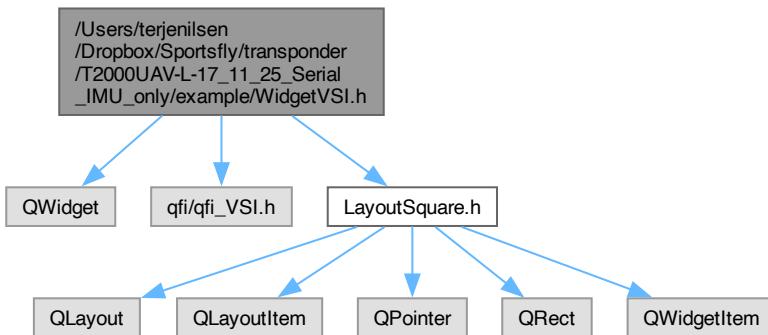
9.28 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/WidgetVSI.cpp File Reference

```
#include <example/WidgetVSI.h>
#include <ui_WidgetVSI.h>
Include dependency graph for WidgetVSI.cpp:
```

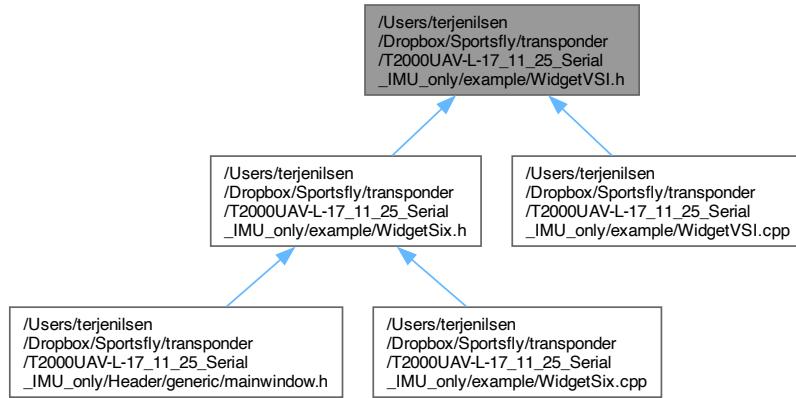


9.29 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/WidgetVSI.h File Reference

```
#include <QWidget>
#include <qfi/qfi_VSI.h>
#include "LayoutSquare.h"
Include dependency graph for WidgetVSI.h:
```



This graph shows which files directly or indirectly include this file:



Data Structures

- class [WidgetVSI](#)

Namespaces

- namespace [Ui](#)

9.30 WidgetVSI.h

[Go to the documentation of this file.](#)

```

00001 /*****  
00002 * Copyright (C) 2021 Marek M. Cel  
00003 *  
00004 * Permission is hereby granted, free of charge, to any person obtaining  
00005 * a copy of this software and associated documentation files (the "Software"),  
00006 * to deal in the Software without restriction, including without limitation  
00007 * the rights to use, copy, modify, merge, publish, distribute, sublicense,  
00008 * and/or sell copies of the Software, and to permit persons to whom  
00009 * the Software is furnished to do so, subject to the following conditions:  
00010 *  
00011 * The above copyright notice and this permission notice shall be included  
00012 * in all copies or substantial portions of the Software.  
00013 *  
00014 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS  
00015 * OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,  
00016 * FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL  
00017 * THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER  
00018 * LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,  
00019 * OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS  
00020 * IN THE SOFTWARE.  
00021 *****  
00022 #ifndef WIDGETVSI_H  
00023 #define WIDGETVSI_H  
00024  
00026  
00027 #include <QWidget>  
00028  
00029 #include <qfi/qfi_VSI.h>  
00030  
00031 #include "LayoutSquare.h"  
00032  
00034

```

```

00035 namespace Ui
00036 {
00037     class WidgetVSI;
00038 }
00039
00041
00042 class WidgetVSI : public QWidget
00043 {
00044     Q_OBJECT
00045
00046 public:
00047
00048     explicit WidgetVSI( QWidget *parent = Q_NULLPTR );
00049
00050     ~WidgetVSI();
00051
00052     inline void redraw() { _vsi->redraw(); }
00053     inline void reinit() { _vsi->reinit(); }
00054
00055     inline void setClimbRate( double climbRate )
00056     {
00057         _vsi->setClimbRate( climbRate );
00058     }
00059
00060 private:
00061
00062     Ui::WidgetVSI *_ui;
00063     qfi_VSI      *_vsi;
00064     LayoutSquare *_layoutSq;
00065
00066     void setupUi();
00067 };
00068
00069
00070
00071 #endif // WIDGETVSI_H

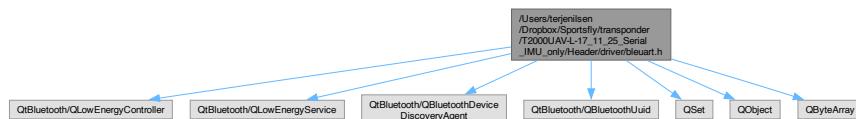
```

9.31 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_← 11_25_Serial_IMU_only/Header/driver/bleuart.h File Reference

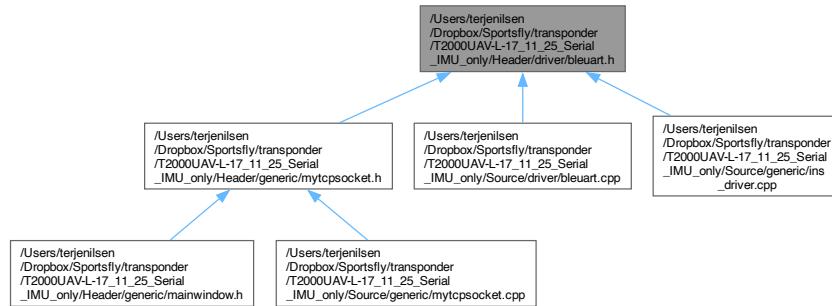
```

#include <QtBluetooth/QLowEnergyController>
#include <QtBluetooth/QLowEnergyService>
#include <QtBluetooth/QBluetoothDeviceDiscoveryAgent>
#include <QtBluetooth/QBluetoothUuid>
#include <QSet>
#include <QObject>
#include <QByteArray>
Include dependency graph for bleuart.h:

```



This graph shows which files directly or indirectly include this file:



Data Structures

- class [BleUart](#)
- class [ComBt](#)

Functions

- static const QBluetoothUuid [wt901Service](#) ("{0000ffe5-0000-1000-8000-00805f9a34fb}")
- static const QBluetoothUuid [wt901Rx](#) ("{0000ffe9-0000-1000-8000-00805f9a34fb}")
- static const QBluetoothUuid [wt901Tx](#) ("{0000ffe4-0000-1000-8000-00805f9a34fb}")

9.31.1 Function Documentation

9.31.1.1 [wt901Rx\(\)](#)

```
const QBluetoothUuid wt901Rx (
    "{0000ffe9-0000-1000-8000-00805f9a34fb}" ) [static]
```

9.31.1.2 [wt901Service\(\)](#)

```
const QBluetoothUuid wt901Service (
    "{0000ffe5-0000-1000-8000-00805f9a34fb}" ) [static]
```

Here is the caller graph for this function:



9.31.1.3 wt901Tx()

```
const QBluetoothUuid wt901Tx (
    "{0000ffe4-0000-1000-8000-00805f9a34fb}" ) [static]
```

9.32 bleuart.h

[Go to the documentation of this file.](#)

```
00001 #ifndef BLEUART_H
00002 #define BLEUART_H
00003
00004 #include <QtBluetooth/QLowEnergyController>
00005 #include <QtBluetooth/QLowEnergyService>
00006 #include <QtBluetooth/QBluetoothDeviceDiscoveryAgent>
00007 #include <QtBluetooth/QBluetoothUuid>
00008 #include <QSet> // <-- add this
00009 #include <QObject>
00010 #include <QByteArray>
00011
00012 //static const QBluetoothUuid kNusService("{6E400001-B5A3-F393-E0A9-E50E24DCCA9E}");
00013 //static const QBluetoothUuid kNusRx("{6E400002-B5A3-F393-E0A9-E50E24DCCA9E}"); // Write
00014 //static const QBluetoothUuid kNusTx("{6E400003-B5A3-F393-E0A9-E50E24DCCA9E}"); // Notify
00015
00016
00017 static const QBluetoothUuid wt901Service("{0000ffe5-0000-1000-8000-00805f9a34fb}");
00018 static const QBluetoothUuid wt901Rx("{0000ffe9-0000-1000-8000-00805f9a34fb}"); // Write
00019 static const QBluetoothUuid wt901Tx("{0000ffe4-0000-1000-8000-00805f9a34fb}"); // Notify
00020 // f00e1ecb-31b9-c6e7-9633-63d3ea57e7e0
00021
00022 class BleUart : public QObject {
00023     Q_OBJECT
00024 public:
00025     bool scancomplete = false;
00026
00027     void startScan(int timeoutMs = 5000, int rssiMin = -95) {
00028         scancomplete = false;
00029
00030         qDebug() << "Scanning BLE (no uid filter in Qt6 dev-scan) for up to" << timeoutMs << "ms...";
00031         candidates.clear();
00032         seen.clear();
00033         targetSvc = QBluetoothUuid();
00034         if (agent) { agent->deleteLater(); agent = nullptr; }
00035
00036         agent = new QBluetoothDeviceDiscoveryAgent(this);
00037         agent->setLowEnergyDiscoveryTimeout(timeoutMs);
00038         connect(agent, &QBluetoothDeviceDiscoveryAgent::deviceDiscovered, this, &BleUart::onDevice);
00039         connect(agent, &QBluetoothDeviceDiscoveryAgent::finished, this, &BleUart::onScanDone);
00040
00041         minRssi = rssiMin;
00042         agent->start(QBluetoothDeviceDiscoveryAgent::LowEnergyMethod);
00043     }
00044 }
00045
00046 void connectTo(const QBluetoothDeviceInfo &dev) {
00047     ctl = QLowEnergyController::createCentral(dev, this);
00048     connect(ctl, &QLowEnergyController::connected, this, &BleUart::onConnected);
00049     connect(ctl, &QLowEnergyController::disconnected, this, &BleUart::onDisconnected);
00050     connect(ctl, &QLowEnergyController::serviceDiscovered, this, &BleUart::onServiceFound);
00051     connect(ctl, &QLowEnergyController::discoveryFinished, this, &BleUart::onServiceScanDone);
00052     connect(ctl, &QLowEnergyController::mtuChanged, this, [this](int mtu){
00053         payloadBytes = qMax(20, mtu - 3); // ATT header = 3 bytes
00054     });
00055     ctl->connectToDevice();
00056 }
00057
00058 void writeBytes(const QByteArray &data) {
00059     if (!svc || !rx.isValid()) return;
00060     for (int i = 0; i < data.size(); i += payloadBytes) {
00061         const int n = qMin(payloadBytes, data.size() - i);
00062         svc->writeCharacteristic(rx, data.mid(i, n),
00063                                     QLowEnergyService::WriteWithoutResponse);
00064     }
00065 }
00066
00067 signals:
00068     void bytesReceived(QByteArray);
00069     void ready(); // emitted when NUS service is discovered & CCCD enabled
00070     void disconnected(); // emitted when link drops
```

```

00071     void dataReceived(QByteArray);
00072
00073 private slots:
00074 private slots:
00075     void onDevice(const QBluetoothDeviceInfo &d) {
00076         if (!(d.coreConfigurations() & QBluetoothDeviceInfo::LowEnergyCoreConfiguration))
00077             return;
00078
00079         // 1) Skip "no ID" devices (no name shown by the OS/stack)
00080         const QString name = d.name().trimmed();
00081         if (name.isEmpty())
00082             return;
00083
00084         // 2) De-duplicate by deviceUuid (macOS reports 00:00:00:00:00:00 for address)
00085         const QUuid id = d.deviceUuid();
00086         if (seen.contains(id))
00087             return;
00088
00089         // Optional: quick RSSI gate
00090         //    if (d.rssi() != 0 && d.rssi() < minRssi) return;
00091
00092         // 3) Keep only NUS advertisers OR specific whitelisted names
00093         const auto advUuids = d.serviceUuids();
00094         const bool hasNus = advUuids.contains(wt901Service);
00095         const bool hasNus = advUuids.contains(kNusService);
00096         static const QStringList targets = { "ESP32-NUS", "T2000-transponder", "WT901BLE67" };
00097         qDebug() << ">>>>>>>> Found:" << name << d.address().toString() << d.deviceUuid().toString();
00098
00099         if (hasNus || targets.contains(name)) {
00100             seen.insert(id);
00101             qDebug() << ">>>>>>>> Target:" << name << d.address().toString() << d.deviceUuid().toString();
00102             candidates << d;
00103
00104             //stop discovery; this will cause QBluetoothDeviceDiscoveryAgent::finished()
00105             // to be emitted (so onScanDone() will run).
00106             // 1) Prefer abort() for immediate, forceful stop:
00107             if (agent && agent->isActive()) {
00108                 agent->finished(); // usually immediate
00109                 qDebug() << "The scan should stop now...";
00110             }
00111         } else {
00112             // Not a match → ignore
00113             // qDebug() << "Ignoring:" << name;
00114         }
00115     }
00116
00117     void onScanDone() {
00118         agent->stop();
00119         qDebug() << "Scan complete. Found" << candidates.size() << "targets.";
00120         scancomplete = true;
00121         if(candidates.length() == 0) scancomplete = true;
00122         for (const auto &dev : candidates){
00123             connectTo(dev); // optionally connect to each
00124         }
00125     }
00126     void onConnected() {
00127         ctl->discoverServices();
00128     }
00129     void onServiceFound(const QBluetoothUuid &uuid) {
00130         if (uuid == wt901Service) targetSvc = uuid;
00131         if (uuid == kNusService) targetSvc = uuid;
00132     }
00133     void onServiceScanDone() {
00134         if (targetSvc.isNull()) {
00135             // Verify NUS after connect → discoverServices() has finished
00136             if (ctl->services().contains(wt901Service)) targetSvc = wt901Service;
00137         }
00138         if (!targetSvc.isNull()) {
00139             svc = ctl->createServiceObject(targetSvc, this);
00140             if (!svc) return;
00141             connect(svc, &QLowEnergyService::stateChanged, this, &BleUart::onSvcState);
00142             connect(svc, &QLowEnergyService::characteristicChanged, this,
00143                     [this](const QLowEnergyCharacteristic &c, const QByteArray &v){
00144                         if (c.uuid() == wt901Tx) emit bytesReceived(v);
00145                     });
00146             svc->discoverDetails();
00147         } else {
00148             qDebug() << "Connected device has no NUS → disconnecting";
00149             ctl->disconnectFromDevice();
00150         }
00151     }
00152     void onSvcState(QLowEnergyService::ServiceState s) {
00153         if (s != QLowEnergyService::ServiceDiscovered) return;
00154         rx = svc->characteristic(wt901Rx);
00155         tx = svc->characteristic(wt901Tx);
00156
00157         // Enable notifications (Qt 6 enum location)

```

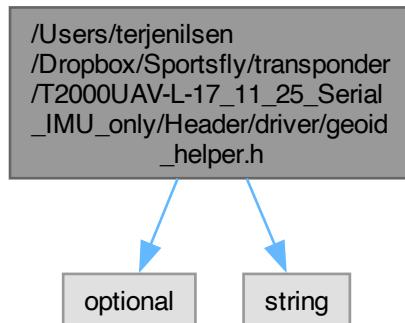
```

00158     auto cccd = tx.descriptor(QBluetoothUuid(
00159         QBluetoothUuid::DescriptorType::ClientCharacteristicConfiguration));
00160     if (cccd.isValid())
00161         svc->writeDescriptor(cccd, QByteArray::fromHex("0100"));
00162
00163     emit ready(); // <-- tell ComBt we're good to send
00164
00165 }
00166 void onDisconnected() {
00167     emit disconnected(); // <-- tell ComBt we're not connected anymore
00168 }
00169
00170 public:
00171     using RxCallback = void*(void *handler, const char* data, uint32_t length); //char*, uint32_t);
00172     void setRxCallback(RxCallback cb) { callback_ = cb; }
00173
00174 private:
00175     // RxCallback callback_ = nullptr;
00176     std::function<void(void *, const char*, uint32_t)> callback_;
00177
00178
00179     QBluetoothDeviceDiscoveryAgent *agent = nullptr;
00180     QList<QBluetoothDeviceInfo> candidates;
00181     QSet<QUuid> seen; // <-- add this to avoid duplicates
00182     QLowEnergyController *ctl = nullptr;
00183     QBluetoothUuid targetSvc;
00184     QLowEnergyService *svc = nullptr;
00185     QBluetoothDeviceInfo currentDev; // optional if you want to track
00186     QLowEnergyCharacteristic rx, tx;
00187     int payloadBytes = 20;
00188     int minRssi = -95;
00189 };
00190
00191 //class BleUart; // fwd
00192
00193 class ComBt : public QObject
00194 {
00195     Q_OBJECT
00196 public:
00197     explicit ComBt(QObject *parent = nullptr);
00198     ~ComBt();
00199
00200     BleUart *serial_ = nullptr;
00201
00202     // BLE "open" - kicks off scan/connect via BleUart, ignores portName/baudrate
00203     bool open(const QString &portName = QString(), qint32 baudrate = 0);
00204     void close();
00205
00206     bool setBaudrate(qint32) { return true; } // not applicable for BLE; always OK
00207
00208     bool send(const QByteArray &data);
00209     bool send(const char *data, unsigned short len) { return send(QByteArray::fromRawData(data, len)); }
00210
00211     bool scanDone = false;
00212     const QString status;
00213     // If you use a C-style callback:
00214     // EXACT same signature for both typedef and std::function
00215     using RxCallback = void*(void* handler, const char* data, uint32_t length);
00216
00217     std::function<void(void *, const char*, uint32_t)> callback_;
00218     void setRxCallback(RxCallback cb) {
00219         serial_->setRxCallback(cb);
00220         callback_ = cb;
00221     }
00222
00223     QObject* parent;
00224
00225 signals:
00226     void connectionChanged(bool connected);
00227     void dataReceived(const QByteArray &data);
00228     void errorReceived(const QString &message);
00229
00230 private:
00231     bool open_ = true;
00232     bool running;
00233     ComBt* g_comqt = nullptr; // simple singleton-style forward -- adjust to your needs
00234
00235
00236 private slots:
00237     void handleReadyRead();
00238 };
00239
00240 #endif // BLEUART_H

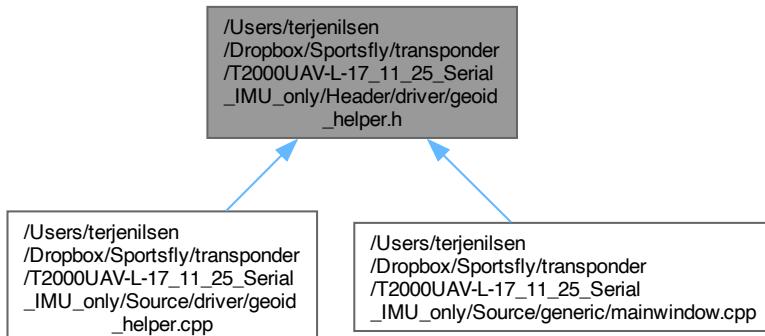
```

9.33 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header/driver/geoid_helper.h File Reference

```
#include <optional>
#include <string>
Include dependency graph for geoid_helper.h:
```



This graph shows which files directly or indirectly include this file:



Data Structures

- struct [GeoidResult](#)
- class [GeoidHelper](#)

9.34 geoid_helper.h

[Go to the documentation of this file.](#)

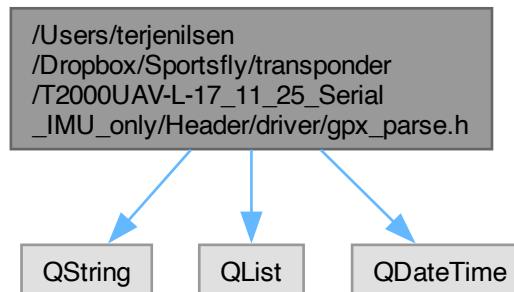
```

00001 #ifndef GEOID_HELPER_H
00002 #define GEOID_HELPER_H
00003
00004 // Same as abou, will fix later...
00005 #pragma once
00006 #include <optional>
00007 #include <string>
00008
00009 struct GeoidResult {
00010     double N;           // geoid separation (meters)
00011     double h_compensated; // h_ellipsoid - alpha * N
00012 };
00013
00014 class GeoidHelper {
00015 public:
00016     // Initialize with path to geoid grid directory or "" to use default search
00017     GeoidHelper(const std::string& geoidDataPath = "");
00018     // alpha: 1.0 -> full AMSL (h - N), 0.0 -> no compensation
00019     std::optional<GeoidResult> compensatedHeight(double lat_deg, double lon_deg,
00020                                                 double h_ellipsoid_m,
00021                                                 double alpha = 1.0);
00022     bool isValid() const;
00023 private:
00024     struct Impl;
00025     Impl* impl = nullptr;
00026 };
00027
00028 #endif // GEOID_HELPER_H

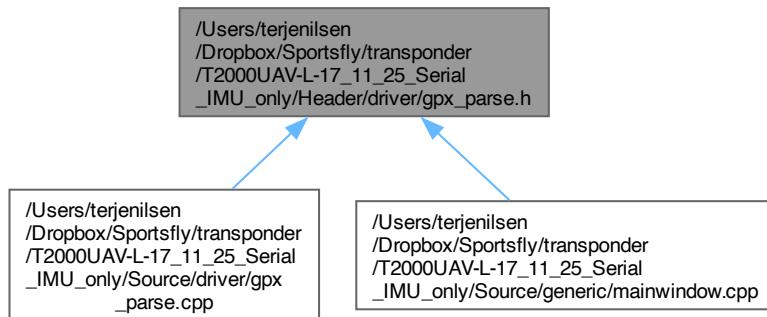
```

9.35 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header/driver/gpx_parse.h File Reference

```
#include <QString>
#include <QList>
#include <QDateTime>
Include dependency graph for gpx_parse.h:
```



This graph shows which files directly or indirectly include this file:



Data Structures

- struct [TrackPoint](#)
- class [GpxParser](#)

9.36 gpx_parse.h

[Go to the documentation of this file.](#)

```

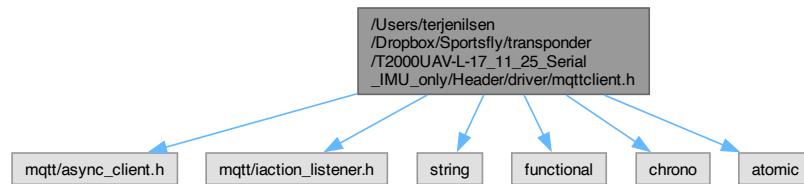
00001 // GPX Used for simulation of GPS data, for testing.
00002
00003 #ifndef GPXPARSER_H
00004 #define GPXPARSER_H
00005
00006 #include <QString>
00007 #include <QList>
00008 #include <QDateTime>
00009 #include <QList>
00010
00011 struct TrackPoint {
00012     double latitude;
00013     double longitude;
00014     double elevation;
00015     double dt;
00016     QDateTime time;
00017     double speed = 0.0; // meters per second
00018 };
00019
00020 class GpxParser {
00021 public:
00022     bool parseFile(const QString &filePath);
00023     QList<TrackPoint> getTrackPoints() const;
00024
00025 private:
00026     QList<TrackPoint> trackPoints;
00027 };
00028
00029 #endif // GPXPARSER_H

```

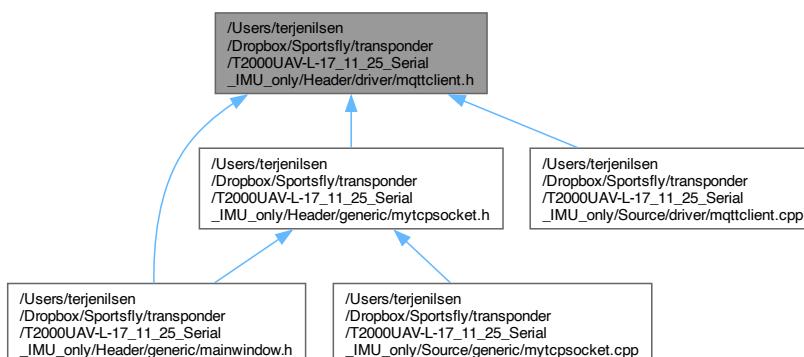
9.37 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header/driver/mqttclient.h File Reference

```
#include <mqtt/async_client.h>
#include <mqtt/iaction_listener.h>
```

```
#include <string>
#include <functional>
#include <chrono>
#include <atomic>
Include dependency graph for mqttclient.h:
```



This graph shows which files directly or indirectly include this file:



Data Structures

- class [ConnListener](#)
Simple connection listener for connect() operations.
- class [MqttClient](#)
Wrapper around mqtt::async_client for "v3-style" MQTT usage.

9.38 mqttclient.h

[Go to the documentation of this file.](#)

```

00001 #ifndef MQTTCLIENT_H
00002 #define MQTTCLIENT_H
00003
00004 // Wrapper around Paho MQTT C++ async_client.
00005 // This header supports two modes:
00006 // - V5 defined: use v5-capable build (but still via async_client).
00007 // - V5 not defined: "v3" style connect/publish/subscribe without v5 properties.
00008
00009 // ps aux | grep mosquitto
  
```

```
00010 // brew services start mosquitto
00011 // mosquitto_pub -t xplane/rpm -m "0.42" -q 1 -r
00012
00013
00014 #include <mqtt/async_client.h>
00015 #include <mqtt/iaction_listener.h> // used in the non-V5 branch
00016 #include <string>
00017 #include <functional>
00018 #include <chrono>
00019 #include <atomic>
00020
00021 #ifdef V5
00022
00023 class MqttClient
00024 {
00025 public:
00026     MqttClient(const std::string& server, const std::string& clientId);
00027     void connect();
00028     void disconnect();
00029     void sendMessage(const std::string& topic, const std::string& message);
00030     void subscribe(const std::string& topic);
00031     void handleIncoming(const std::string& topic, const std::string& payload);
00032     void setMessageHandler(std::function<void(const std::string&, const std::string&)> handler)
00033     {
00034         userMessageHandler_ = std::move(handler);
00035     }
00036     mqtt::async_client& client() { return client_; }
00037
00038 private:
00039     class Callback : public virtual mqtt::callback
00040     {
00041     public:
00042         explicit Callback(MqttClient* owner) : owner_(owner) {}
00043         void message_arrived(mqtt::const_message_ptr msg) override;
00044         void connection_lost(const std::string& cause) override;
00045
00046     private:
00047         MqttClient* owner_;
00048     };
00049     std::function<void(const std::string&, const std::string&)> userMessageHandler_;
00050     mqtt::async_client client_;
00051     Callback cb_;
00052 };
00053
00054 #else // ----- non-V5 branch -----
00055
00056 class MqttClient;
00057
00058 class ConnListener : public virtual mqtt::iaction_listener
00059 {
00060 public:
00061     explicit ConnListener(MqttClient* o) : owner(o) {}
00062     void on_failure(const mqtt::token& tok) override;
00063     void on_success(const mqtt::token& tok) override;
00064
00065 private:
00066     MqttClient* owner = nullptr;
00067 };
00068
00069 class MqttClient
00070 {
00071 public:
00072     using MsgHandler = std::function<void(const std::string&, const std::string&)>;
00073     MqttClient(const std::string& server, const std::string& clientId);
00074     void connect(bool cleanSession = true,
00075                  std::chrono::seconds keepAlive = std::chrono::seconds(20));
00076     void disconnect();
00077     void sendMessage(const std::string& topic,
00078                      const std::string& payload,
00079                      int qos = 1,
00080                      bool retained = false);
00081
00082
00083
00084
00085
00086
00087
00088
00089
00090
00091
00092
00093
00094
00095
00096
00097
00098
00099
00100
00101
00102
00103
00104
00105
00106
00107
00108
00109
00110
00111
00112
00113
00114
00115
00116
00117
00118
00119
00120
00121
00122
00123
00124
00125
00126
00127
00128
00129
00130
00131
00132
00133
00134
00135
00136
00137
00138
00139
00140
00141
00142
00143
00144
00145
00146
00147
00148
00149
00150
00151
00152
00153
00154
00155
00156
00157
00158
00159
00160
00161
00162
00163
00164
00165
00166
00167
00168
00169
00170
00171
00172
00173
00174
00175
00176
00177
00178
00179
00180
00181
00182
00183
00184
00185
00186
00187
00188
00189
00190
00191
00192
00193
00194
00195
00196
00197
00198
00199
00200
00201
00202
00203
00204
00205
00206
00207
00208
00209
00210
00211
00212
00213
00214
00215
00216
00217
00218
00219
00220
00221
00222
00223
00224
00225
00226
00227
00228
00229
00230
00231
00232
00233
00234
00235
00236
00237
00238
00239
00240
00241
00242
00243
00244
00245
00246
00247
00248
00249
00250
00251
00252
00253
00254
00255
00256
00257
00258
00259
00260
00261
00262
00263
00264
00265
00266
00267
00268
00269
00270
00271
00272
00273
00274
00275
00276
00277
00278
00279
00280
00281
00282
00283
00284
00285
00286
00287
00288
00289
00290
00291
00292
00293
00294
00295
00296
00297
00298
00299
00300
00301
00302
00303
00304
00305
00306
00307
00308
00309
00310
00311
00312
00313
00314
00315
00316
00317
00318
00319
00320
00321
00322
00323
00324
00325
00326
00327
00328
00329
00330
00331
00332
00333
00334
00335
00336
00337
00338
00339
00340
00341
00342
00343
00344
00345
00346
00347
00348
00349
00350
00351
00352
00353
00354
00355
00356
00357
00358
00359
00360
00361
00362
00363
00364
00365
00366
00367
00368
00369
00370
00371
00372
00373
00374
00375
00376
00377
00378
00379
00380
00381
00382
00383
00384
00385
00386
00387
00388
00389
00390
00391
00392
00393
00394
00395
00396
00397
00398
00399
00400
00401
00402
00403
00404
00405
00406
00407
00408
00409
00410
00411
00412
00413
00414
00415
00416
00417
00418
00419
00420
00421
00422
00423
00424
00425
00426
00427
00428
00429
00430
00431
00432
00433
00434
00435
00436
00437
00438
00439
00440
00441
00442
00443
00444
00445
00446
00447
00448
00449
00450
00451
00452
00453
00454
00455
00456
00457
00458
00459
00460
00461
00462
00463
00464
00465
00466
00467
00468
00469
00470
00471
00472
00473
00474
00475
00476
00477
00478
00479
00480
00481
00482
00483
00484
00485
00486
00487
00488
00489
00490
00491
00492
00493
00494
00495
00496
00497
00498
00499
00500
00501
00502
00503
00504
00505
00506
00507
00508
00509
00510
00511
00512
00513
00514
00515
00516
00517
00518
00519
00520
00521
00522
00523
00524
00525
00526
00527
00528
00529
00530
00531
00532
00533
00534
00535
00536
00537
00538
00539
00540
00541
00542
00543
00544
00545
00546
00547
00548
00549
00550
00551
00552
00553
00554
00555
00556
00557
00558
00559
00560
00561
00562
00563
00564
00565
00566
00567
00568
00569
00570
00571
00572
00573
00574
00575
00576
00577
00578
00579
00580
00581
00582
00583
00584
00585
00586
00587
00588
00589
00590
00591
00592
00593
00594
00595
00596
00597
00598
00599
00600
00601
00602
00603
00604
00605
00606
00607
00608
00609
00610
00611
00612
00613
00614
00615
00616
00617
00618
00619
00620
00621
00622
00623
00624
00625
00626
00627
00628
00629
00630
00631
00632
00633
00634
00635
00636
00637
00638
00639
00640
00641
00642
00643
00644
00645
00646
00647
00648
00649
00650
00651
00652
00653
00654
00655
00656
00657
00658
00659
00660
00661
00662
00663
00664
00665
00666
00667
00668
00669
00670
00671
00672
00673
00674
00675
00676
00677
00678
00679
00680
00681
00682
00683
00684
00685
00686
00687
00688
00689
00690
00691
00692
00693
00694
00695
00696
00697
00698
00699
00700
00701
00702
00703
00704
00705
00706
00707
00708
00709
00710
00711
00712
00713
00714
00715
00716
00717
00718
00719
00720
00721
00722
00723
00724
00725
00726
00727
00728
00729
00730
00731
00732
00733
00734
00735
00736
00737
00738
00739
00740
00741
00742
00743
00744
00745
00746
00747
00748
00749
00750
00751
00752
00753
00754
00755
00756
00757
00758
00759
00760
00761
00762
00763
00764
00765
00766
00767
00768
00769
00770
00771
00772
00773
00774
00775
00776
00777
00778
00779
00780
00781
00782
00783
00784
00785
00786
00787
00788
00789
00790
00791
00792
00793
00794
00795
00796
00797
00798
00799
00800
00801
00802
00803
00804
00805
00806
00807
00808
00809
00810
00811
00812
00813
00814
00815
00816
00817
00818
00819
00820
00821
00822
00823
00824
00825
00826
00827
00828
00829
00830
00831
00832
00833
00834
00835
00836
00837
00838
00839
00840
00841
00842
00843
00844
00845
00846
00847
00848
00849
00850
00851
00852
00853
00854
00855
00856
00857
00858
00859
00860
00861
00862
00863
00864
00865
00866
00867
00868
00869
00870
00871
00872
00873
00874
00875
00876
00877
00878
00879
00880
00881
00882
00883
00884
00885
00886
00887
00888
00889
00890
00891
00892
00893
00894
00895
00896
00897
00898
00899
00900
00901
00902
00903
00904
00905
00906
00907
00908
00909
00910
00911
00912
00913
00914
00915
00916
00917
00918
00919
00920
00921
00922
00923
00924
00925
00926
00927
00928
00929
00930
00931
00932
00933
00934
00935
00936
00937
00938
00939
00940
00941
00942
00943
00944
00945
00946
00947
00948
00949
00950
00951
00952
00953
00954
00955
00956
00957
00958
00959
00960
00961
00962
00963
00964
00965
00966
00967
00968
00969
00970
00971
00972
00973
00974
00975
00976
00977
00978
00979
00980
00981
00982
00983
00984
00985
00986
00987
00988
00989
00990
00991
00992
00993
00994
00995
00996
00997
00998
00999
00999
```

```

00191
00198     void subscribe(const std::string& topic, int qos = 1);
00199
00205     void setMessageHandler(MsgHandler handler) { userMessageHandler_ = std::move(handler); }
00206
00210     mqtt::async_client& client() { return client_; }
00211
00212     private:
00216         class Callback : public virtual mqtt::callback
00217     {
00218     public:
00219         explicit Callback(MqttClient* owner) : owner_(owner) {}
00220
00221         void message_arrived(mqtt::const_message_ptr msg) override;
00222         void connection_lost(const std::string& cause) override;
00223
00224     private:
00225         MqttClient* owner_ = nullptr;
00226     };
00227
00229     MsgHandler userMessageHandler_;
00230
00232     mqtt::async_client client_;
00233
00235     Callback cb_;
00236 };
00237
00238 #endif // V5 / !V5
00239
00240 #endif // MQTTCLIENT_H

```

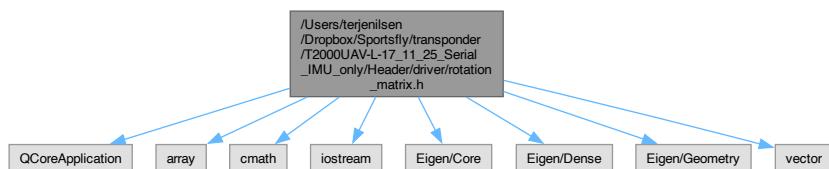
9.39 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header/driver/rotation_matrix.h File Reference

```

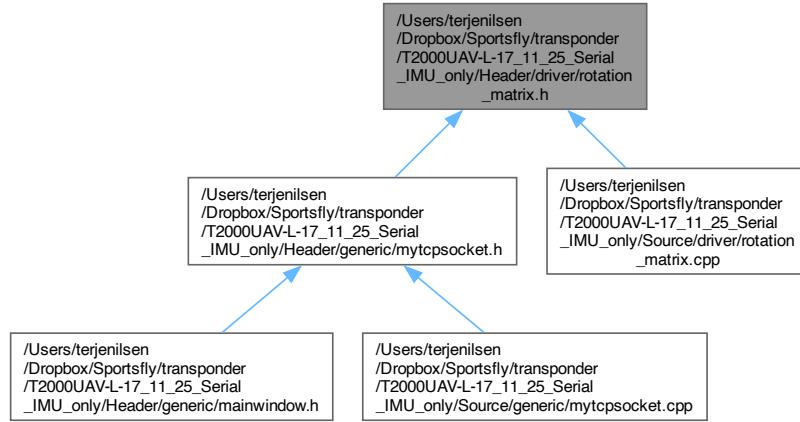
#include <QCoreApplication>
#include <array>
#include <cmath>
#include <iostream>
#include <Eigen/Core>
#include <Eigen/Dense>
#include <Eigen/Geometry>
#include <vector>

```

Include dependency graph for rotation_matrix.h:



This graph shows which files directly or indirectly include this file:



Macros

- `#define DEG_TO_RAD (3.141592/180)`
- `#define RAD_TO_DEG (180.0/3.141592765)`
- `#define M_PI 3.14159265358979323846`

Typedefs

- using `Vector3x` = `std::array<double, 3>`
- using `Vector9` = `std::array<double, 9>`
- using `Matrix3x3` = `std::array<std::array<double, 3>, 3>`
- using `Matrix3x6` = `std::array<std::array<double, 6>, 3>`
- using `Matrix9x9` = `std::array<std::array<double, 9>, 9>`

Functions

- void `printVector` (const `Vector3x` &vector)
- void `rotateSensors` (`Vector3x` &gyro, `Vector3x` &accel, `Vector3x` &mag, double theta)
- void `rotateSensors` (`Vector3x` &gyro, `Vector3x` &accel, `Vector3x` &mag, `Matrix3x3` rotationMatrix)
- `Vector3x rotateVector` (const `Matrix3x3` &rotationMatrix, const `Vector3x` &vector)
- `Matrix3x3 multiplyMatrix` (const `Matrix3x3` &mat1, const `Matrix3x3` &mat2)
- `Matrix3x3 createRotationMatrix` (`Vector3x` theta)
- `Matrix3x3 createRotationMatrixZ` (double theta)
- void `rotateSensors` (`Vector3x` &gyro, `Vector3x` &accel, `Vector3x` &mag, `Vector3x` &attitude, `Matrix3x3` rotationMatrix)

9.39.1 Macro Definition Documentation

9.39.1.1 DEG_TO_RAD

```
#define DEG_TO_RAD (3.141592/180)
```

9.39.1.2 M_PI

```
#define M_PI 3.14159265358979323846
```

9.39.1.3 RAD_TO_DEG

```
#define RAD_TO_DEG (180.0/3.141592765)
```

9.39.2 Typedef Documentation

9.39.2.1 Matrix3x3

```
using Matrix3x3 = std::array<std::array<double, 3>, 3>
```

9.39.2.2 Matrix3x6

```
using Matrix3x6 = std::array<std::array<double, 6>, 3>
```

9.39.2.3 Matrix9x9

```
using Matrix9x9 = std::array<std::array<double, 9>, 9>
```

9.39.2.4 Vector3x

```
using Vector3x = std::array<double, 3>
```

9.39.2.5 Vector9

```
using Vector9 = std::array<double, 9>
```

9.39.3 Function Documentation

9.39.3.1 createRotationMatrix()

```
Matrix3x3 createRotationMatrix (
    Vector3x theta)
```

Here is the call graph for this function:



Here is the caller graph for this function:



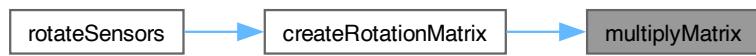
9.39.3.2 createRotationMatrixZ()

```
Matrix3x3 createRotationMatrixZ (
    double theta)
```

9.39.3.3 multiplyMatrix()

```
Matrix3x3 multiplyMatrix (
    const Matrix3x3 & mat1,
    const Matrix3x3 & mat2)
```

Here is the caller graph for this function:



9.39.3.4 printVector()

```
void printVector (
    const Vector3x & vector)
```

9.39.3.5 rotateSensors() [1/3]

```
void rotateSensors (
    Vector3x & gyro,
    Vector3x & accel,
    Vector3x & mag,
    double theta)
```

9.39.3.6 rotateSensors() [2/3]

```
void rotateSensors (
    Vector3x & gyro,
    Vector3x & accel,
    Vector3x & mag,
    Matrix3x3 rotationMatrix)
```

Here is the call graph for this function:



9.39.3.7 rotateSensors() [3/3]

```
void rotateSensors (
    Vector3x & gyro,
    Vector3x & accel,
    Vector3x & mag,
    Vector3x & attitude,
    Matrix3x3 rotationMatrix)
```

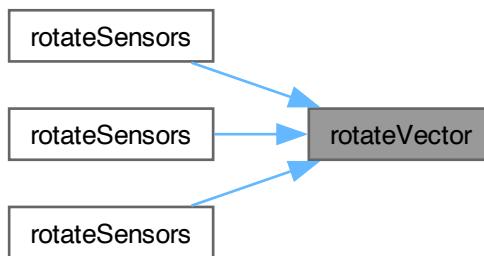
Here is the call graph for this function:



9.39.3.8 rotateVector()

```
Vector3x rotateVector (
    const Matrix3x3 & rotationMatrix,
    const Vector3x & vector)
```

Here is the caller graph for this function:



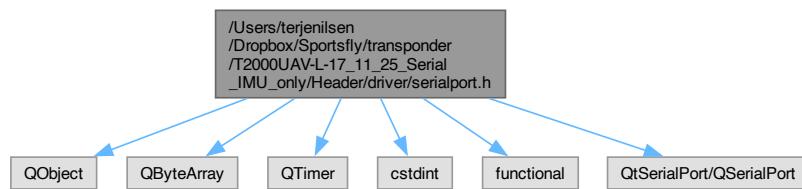
9.40 rotation_matrix.h

[Go to the documentation of this file.](#)

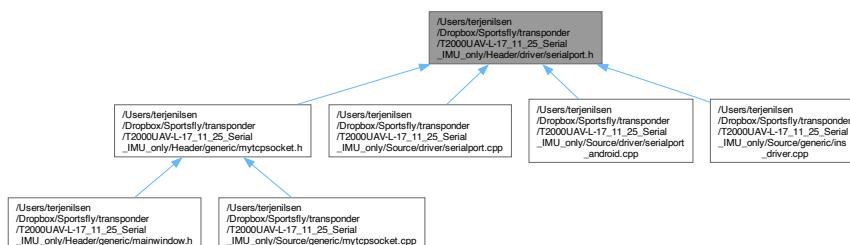
```
00001 #ifndef ROTATION_MATRIX_H
00002 #define ROTATION_MATRIX_H
00003
00004 #include <QCoreApplication>
00005 #include <array>      // For std::array
00006 #include <cmath>       // For sin, cos, M_PI
00007 #include <iostream>   // For std::cout
00008 #include <array>
00009 #include <Eigen/Core>
00010 #include <Eigen/Dense>
00011 #include <Eigen/Geometry> // for AngleAxis
00012 #include <vector>
00013 #include <cmath>
00014
00015 using namespace Eigen;
00016
00017 // Define Vector3x as a 3-element array to represent 3D vectors (x, y, z)
00018 using Vector3x = std::array<double, 3>;
00019 using Vector9 = std::array<double, 9>;
00020
00021 #define DEG_TO_RAD (3.141592/180)
00022 #define RAD_TO_DEG (180.0/3.141592765)
00023
00024 #ifndef M_PI
00025 #define M_PI 3.14159265358979323846
00026 #endif
00027
00028 using Matrix3x3 = std::array<std::array<double, 3>, 3>;
00029 using Matrix3x6 = std::array<std::array<double, 6>, 3>;
00030 using Matrix9x9 = std::array<std::array<double, 9>, 9>;
00031
00032 void printVector(const Vector3x& vector);
00033 void rotateSensors(Vector3x& gyro, Vector3x& accel, Vector3x& mag, double theta);
00034 void rotateSensors(Vector3x& gyro, Vector3x& accel, Vector3x& mag, Matrix3x3 rotationMatrix);
00035
00036 Vector3x rotateVector(const Matrix3x3& rotationMatrix, const Vector3x& vector);
00037 Matrix3x3 multiplyMatrix(const Matrix3x3& mat1, const Matrix3x3& mat2);
00038 Matrix3x3 createRotationMatrix(Vector3x theta);
00039 Matrix3x3 createRotationMatrixZ(double theta);
00040 void rotateSensors(Vector3x& gyro, Vector3x& accel, Vector3x& mag, Vector3x& attitude, Matrix3x3
    rotationMatrix);
00041
00042 #endif // ROTATION_MATRIX_H
```

9.41 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header/driver/serialport.h File Reference

```
#include <QObject>
#include <QByteArray>
#include <QTimer>
#include <cstdint>
#include <functional>
#include <QtSerialPort/QSerialPort>
Include dependency graph for serialport.h:
```



This graph shows which files directly or indirectly include this file:



Data Structures

- class [ComQt](#)
Thin wrapper around QSerialPort used by [MyTcpSocket](#).

9.42 serialport.h

[Go to the documentation of this file.](#)

```
00001 #ifndef COMQT_H
00002 #define COMQT_H
00003
00004 #include <QObject>
00005 #include <QByteArray>
00006 #include <QTimer>
00007 #include <cstdint>
00008 #include <functional>
00009
00010 #ifndef Q_OS_IOS
```

```

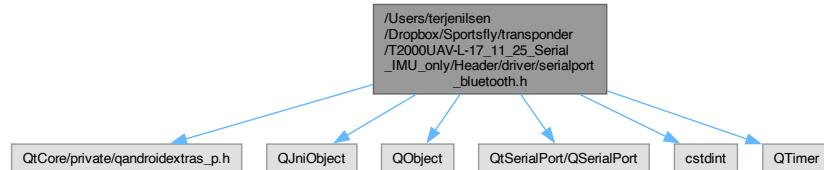
00011 #include <QtSerialPort/QSerialPort>
00012 #endif
00013
00014 #ifdef Q_OS_ANDROID
00015 #include <QtCore/private/qandroidextras_p.h>
00016 #include <QJniObject>
00017 #endif
00018
00019 class ComQt : public QObject
00020 {
00021     Q_OBJECT
00022
00023 public:
00024     explicit ComQt(QObject *parent = nullptr);
00025     ~ComQt();
00026
00027     bool open(const QString &portName, qint32 baudrate);
00028
00029     void close();
00030
00031     bool setBaudrate(qint32 baudrate);
00032
00033     bool send(const QByteArray &data);
00034
00035     bool send(const char *data, unsigned short len);
00036
00037 #ifndef Q_OS_IOS
00038     QSerialPort *serial_ = nullptr;
00039
00040     bool isOpen() const { return serial_ && serial_->isOpen(); }
00041 #endif
00042
00043 signals:
00044     void connectionChanged(bool connected);
00045
00046     void dataReceived(const QByteArray &data);
00047
00048     void errorReceived(const QString &message);
00049
00050 public:
00051     const QString status;
00052
00053     // -----
00054     // C-style RX callback support
00055     // -----
00056
00057     using RxCallback = void*(void *handler, const char *data, uint32_t length);
00058
00059     std::function<void(void *, const char *, uint32_t)> callback_;
00060
00061     void setRxCallback(RxCallback cb) { callback_ = cb; }
00062
00063     QObject *parent = nullptr;
00064
00065 #ifdef Q_OS_ANDROID
00066     QJniObject *someJavaObject = nullptr;
00067     QJniObject *imuJavaObject = nullptr;
00068 #endif
00069
00070 private:
00071     bool running = false;
00072
00073     ComQt *g_comqt = nullptr;
00074
00075 private slots:
00076     void handleReadyRead();
00077 };
00078
00079 #endif // COMQT_H

```

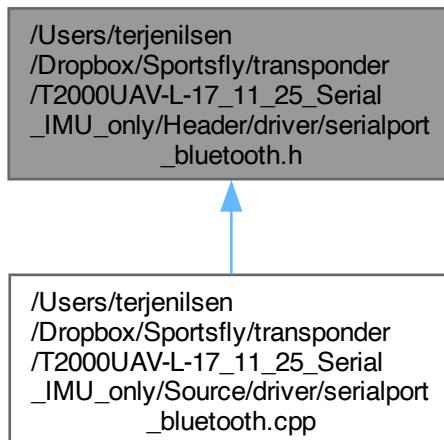
9.43 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_← 11_25_Serial_IMU_only/Header/driver/serialport_bluetooth.h File Reference

```
#include <QtCore/private/qandroidextras_p.h>
#include <QJniObject>
#include <QObject>
```

```
#include <QtSerialPort/QSerialPort>
#include <cstdint>
#include <QTimer>
Include dependency graph for serialport_bluetooth.h:
```



This graph shows which files directly or indirectly include this file:



Data Structures

- class [BTComQt](#)

9.44 serialport_bluetooth.h

[Go to the documentation of this file.](#)

```
00001 #ifndef BTCOMQT_H
00002 #define BTCOMQT_H
00003
00004 #include <QtCore/private/qandroidextras_p.h>
00005 #include <QJniObject>
00006 #include <QObject>
00007 #include <QtSerialPort/QSerialPort>
00008 #include <cstdint>
00009 #include <QTimer>
```

```

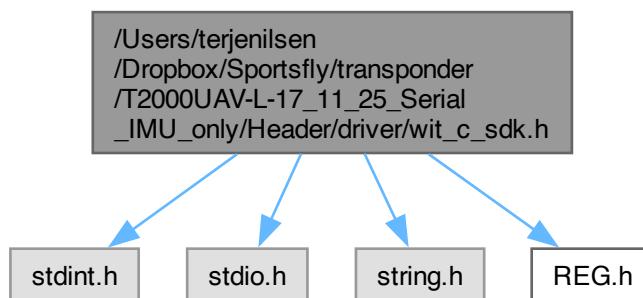
00010
00011 class BTComQt : public QObject
00012 {
00013     Q_OBJECT
00014 public:
00015     explicit BTComQt(QObject *parent = nullptr);
00016     ~BTComQt();
00017
00018     // Open by port name (e.g. "COM4" on Windows, "/dev/tty.usbserial-14130" on macOS)
00019     bool open(const QString &portName, qint32 baudrate);
00020
00021     void close();
00022
00023     bool send(const QByteArray &data);
00024     bool send(const char *data, unsigned short len);
00025
00026     QJniObject *imuJavaObject = nullptr;
00027
00028     using RxCallback = void*(void *handler, const char* data, uint32_t length); //char*, uint32_t);
00029 //     using RxCallback = void*(void *, const QByteArray &data); //char*, uint32_t);
00030     void setRxCallback(RxCallback cb) { callback_ = cb; }
00031
00032 //private:
00033     QSerialPort *serial_ = nullptr;
00034     QTimer* timerAndroid;
00035
00036 private:
00037     RxCallback callback_ = nullptr;
00038     QObject *parent;
00039
00040 signals:
00041     void dataReceived(const QByteArray &data);
00042
00043 private slots:
00044     void handleReadyRead();
00045 };
00046
00047 #endif // BTCOMQT_H

```

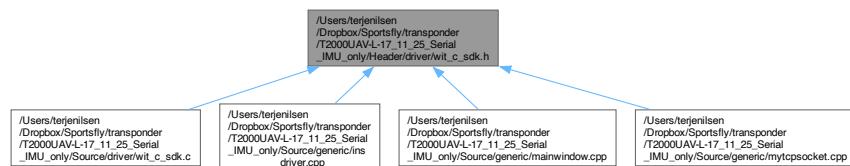
9.45 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_← 11_25_Serial_IMU_only/Header/driver/wit_c_sdk.h File Reference

This file provides all Configure sensor function.

```
#include <stdint.h>
#include <stdio.h>
#include <string.h>
#include "REG.h"
Include dependency graph for wit_c_sdk.h:
```



This graph shows which files directly or indirectly include this file:



Macros

- #define SLEEP_MS(ms)
- #define SCANF_INT(fmt, ptr)
- #define UINT32 uint32_t
- #define WIT_HAL_OK (0)
- #define WIT_HAL_BUSY (-1)
- #define WIT_HAL_TIMEOUT (-2)
- #define WIT_HAL_ERROR (-3)
- #define WIT_HAL_NOMEM (-4)
- #define WIT_HAL_EMPTY (-5)
- #define WIT_HAL_INVAL (-6)
- #define WIT_DATA_BUFF_SIZE 256
- #define WIT_PROTOCOL_NORMAL 0
- #define WIT_PROTOCOL_MODBUS 1
- #define WIT_PROTOCOL_CAN 2
- #define WIT_PROTOCOL_I2C 3
- #define WIT_PROTOCOL_JY61 4
- #define WIT_PROTOCOL_905x_MODBUS 5
- #define WIT_PROTOCOL_905x_CAN 6

Typedefs

- typedef void(* **SerialWrite**) (uint8_t *p_ucData, uint32_t uiLen)
- typedef void(* **DelaymsCb**) (uint16_t ucMs)
- typedef void(* **RegUpdateCb**) (uint32_t uiReg, uint32_t uiRegNum)

Functions

- int32_t **WitSerialWriteRegister** (**SerialWrite** write_func)
- int32_t **WitDelayMsRegister** (**DelaymsCb** delayms_func)
- int32_t **WitRegisterCallBack** (**RegUpdateCb** update_func)
- int32_t **WitWriteReg** (uint32_t uiReg, uint16_t usData)
- int32_t **WitReadReg** (uint32_t uiReg, uint32_t uiReadNum)
- int32_t **WitInit** (uint32_t uiProtocol, uint8_t ucAddr)
- void **WitDeInit** (void)
- void **WitSerialDataIn** (void *handler, const char *data, uint32_t length)
- void **posix_delay_ms** (uint16_t ms)
- int32_t **WitStartAccCali** (void)
- int32_t **WitStopAccCali** (void)
- int32_t **WitStartMagCali** (void)

- int32_t [WitStopMagCali](#) (void)
- int32_t [WitSetUartBaud](#) (int32_t uiBaudIndex)
- int32_t [WitSetBandwidth](#) (int32_t uiBaudWidth)
- int32_t [WitSetOutputRate](#) (int32_t uiRate)
- int32_t [WitSetContent](#) (int32_t uiRsw)
- int32_t [WitSetCanBaud](#) (int32_t uiBaudIndex)
- int32_t [WitSaveParameter](#) (void)
- int32_t [WitSetForReset](#) (void)
- int32_t [WitCaliRefAngle](#) (void)
- char [CheckRange](#) (short sTemp, short sMin, short sMax)

Variables

- int16_t [sReg](#) [REGSIZE]

9.45.1 Detailed Description

This file provides all Configure sensor function.

Author

Wit

Version

V1.0

Date

05-May-2022

Attention

<http://wit-motion.cn/>

9.45.2 Macro Definition Documentation

9.45.2.1 SCANF_INT

```
#define SCANF_INT(  
    fmt,  
    ptr)
```

Value:

scanf(fmt, ptr)

9.45.2.2 SLEEP_MS

```
#define SLEEP_MS(  
    ms)
```

Value:

```
usleep((ms) * 1000)
```

9.45.2.3 UINT32

```
#define UINT32 uint32_t
```

9.45.2.4 WIT_DATA_BUFF_SIZE

```
#define WIT_DATA_BUFF_SIZE 256
```

9.45.2.5 WIT_HAL_BUSY

```
#define WIT_HAL_BUSY (-1)
```

Busy

9.45.2.6 WIT_HAL_EMPTY

```
#define WIT_HAL_EMPTY (-5)
```

The resource is empty

9.45.2.7 WIT_HAL_ERROR

```
#define WIT_HAL_ERROR (-3)
```

A generic error happens

9.45.2.8 WIT_HAL_INVAL

```
#define WIT_HAL_INVAL (-6)
```

Invalid argument

9.45.2.9 WIT_HAL_NOMEM

```
#define WIT_HAL_NOMEM (-4)
```

No memory

9.45.2.10 WIT_HAL_OK

```
#define WIT_HAL_OK (0)
```

There is no error

9.45.2.11 WIT_HAL_TIMEOUT

```
#define WIT_HAL_TIMEOUT (-2)
```

Timed out

9.45.2.12 WIT_PROTOCOL_905x_CAN

```
#define WIT_PROTOCOL_905x_CAN 6
```

9.45.2.13 WIT_PROTOCOL_905x_MODBUS

```
#define WIT_PROTOCOL_905x_MODBUS 5
```

9.45.2.14 WIT_PROTOCOL_CAN

```
#define WIT_PROTOCOL_CAN 2
```

9.45.2.15 WIT_PROTOCOL_I2C

```
#define WIT_PROTOCOL_I2C 3
```

9.45.2.16 WIT_PROTOCOL_JY61

```
#define WIT_PROTOCOL_JY61 4
```

9.45.2.17 WIT_PROTOCOL_MODBUS

```
#define WIT_PROTOCOL_MODBUS 1
```

9.45.2.18 WIT_PROTOCOL_NORMAL

```
#define WIT_PROTOCOL_NORMAL 0
```

9.45.3 Typedef Documentation

9.45.3.1 DelaymsCb

```
typedef void(* DelaymsCb) (uint16_t ucMs)
```

9.45.3.2 RegUpdateCb

```
typedef void(* RegUpdateCb) (uint32_t uiReg, uint32_t uiRegNum)
```

9.45.3.3 SerialWrite

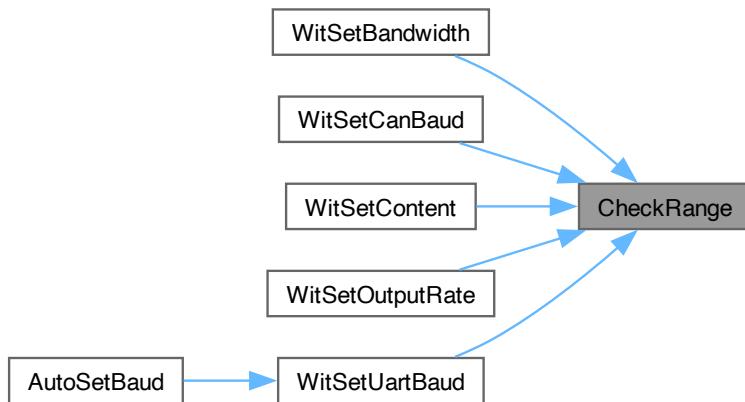
```
typedef void(* SerialWrite) (uint8_t *p_ucData, uint32_t uiLen)
```

9.45.4 Function Documentation

9.45.4.1 CheckRange()

```
char CheckRange (
    short sTemp,
    short sMin,
    short sMax)
```

Here is the caller graph for this function:



9.45.4.2 posix_delay_ms()

```
void posix_delay_ms (
    uint16_t ms)
```

Here is the caller graph for this function:



9.45.4.3 WitCaliRefAngle()

```
int32_t WitCaliRefAngle (
    void )
```

Here is the call graph for this function:



9.45.4.4 WitDeInit()

```
void WitDeInit (
    void )
```

9.45.4.5 WitDelayMsRegister()

```
int32_t WitDelayMsRegister (
    DelaymsCb delayms_func)
```

Here is the caller graph for this function:



9.45.4.6 WitInit()

```
int32_t WitInit (
    uint32_t uiProtocol,
    uint8_t ucAddr)
```

Here is the caller graph for this function:



9.45.4.7 WitReadReg()

```
int32_t WitReadReg (
    uint32_t uiReg,
    uint32_t uiReadNum)
```

Here is the caller graph for this function:



9.45.4.8 WitRegisterCallBack()

```
int32_t WitRegisterCallBack (
    RegUpdateCb update_func)
```

Here is the caller graph for this function:



9.45.4.9 WitSaveParameter()

```
int32_t WitSaveParameter (
    void )
```

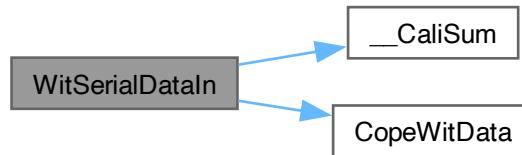
Here is the call graph for this function:



9.45.4.10 WitSerialDataIn()

```
void WitSerialDataIn (
    void * handler,
    const char * data,
    uint32_t length)
```

Here is the call graph for this function:



Here is the caller graph for this function:



9.45.4.11 WitSerialWriteRegister()

```
int32_t WitSerialWriteRegister (
    SerialWrite write_func)
```

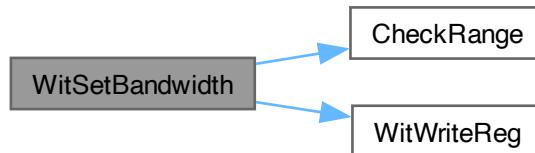
Here is the caller graph for this function:



9.45.4.12 WitSetBandwidth()

```
int32_t WitSetBandwidth (
    int32_t uiBaudWidth)
```

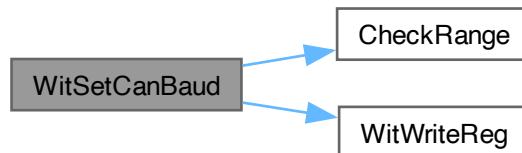
Here is the call graph for this function:



9.45.4.13 WitSetCanBaud()

```
int32_t WitSetCanBaud (
    int32_t uiBaudIndex)
```

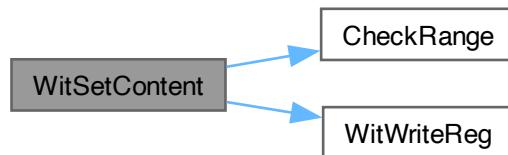
Here is the call graph for this function:



9.45.4.14 WitSetContent()

```
int32_t WitSetContent (
    int32_t uiRsw)
```

Here is the call graph for this function:



9.45.4.15 WitSetForReset()

```
int32_t WitSetForReset (
    void )
```

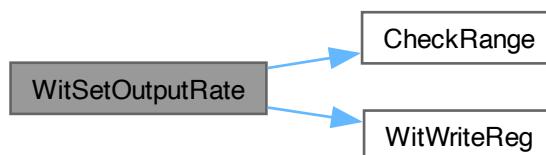
Here is the call graph for this function:



9.45.4.16 WitSetOutputRate()

```
int32_t WitSetOutputRate (
    int32_t uiRate)
```

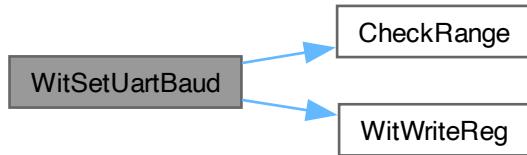
Here is the call graph for this function:



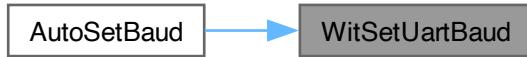
9.45.4.17 WitSetUartBaud()

```
int32_t WitSetUartBaud (
    int32_t uiBaudIndex)
```

Here is the call graph for this function:



Here is the caller graph for this function:



9.45.4.18 WitStartAccCali()

```
int32_t WitStartAccCali (
    void )
```

Here is the call graph for this function:



9.45.4.19 WitStartMagCali()

```
int32_t WitStartMagCali (
    void )
```

Here is the call graph for this function:



9.45.4.20 WitStopAccCali()

```
int32_t WitStopAccCali (
    void )
```

Here is the call graph for this function:



9.45.4.21 WitStopMagCali()

```
int32_t WitStopMagCali (
    void )
```

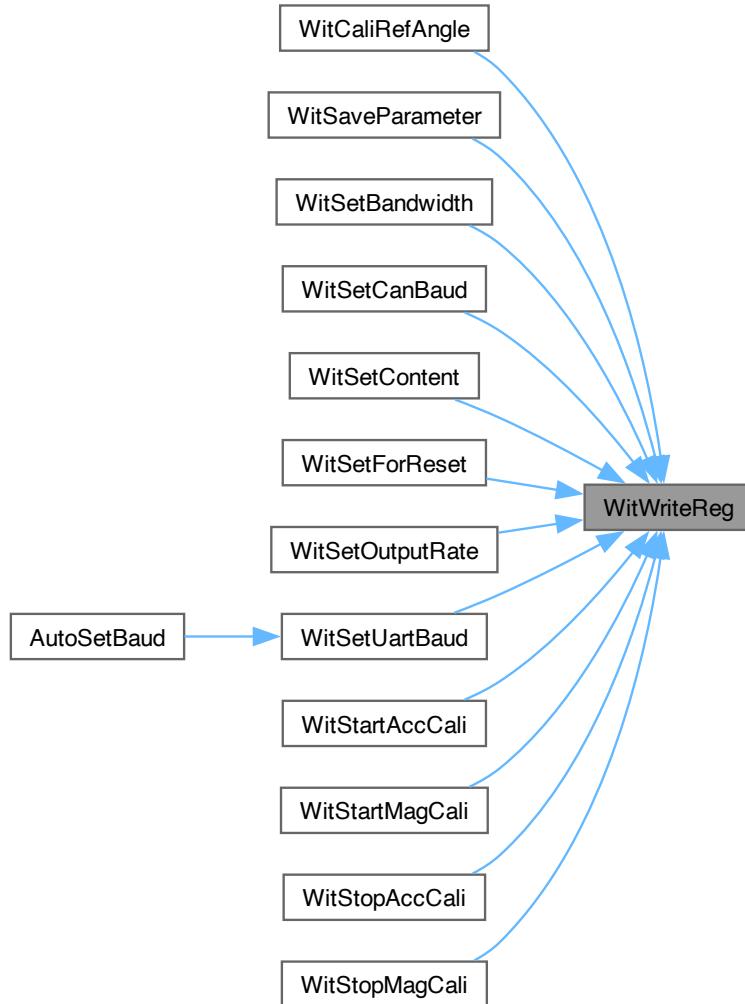
Here is the call graph for this function:



9.45.4.22 WitWriteReg()

```
int32_t WitWriteReg (
    uint32_t uiReg,
    uint16_t usData)
```

Here is the caller graph for this function:



9.45.5 Variable Documentation

9.45.5.1 sReg

```
int16_t sReg[REGSIZE] [extern]
```

9.46 wit_c_sdk.h

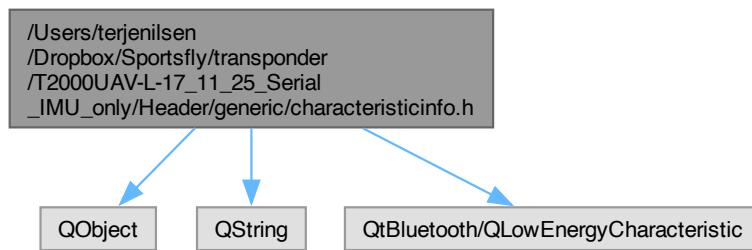
[Go to the documentation of this file.](#)

```

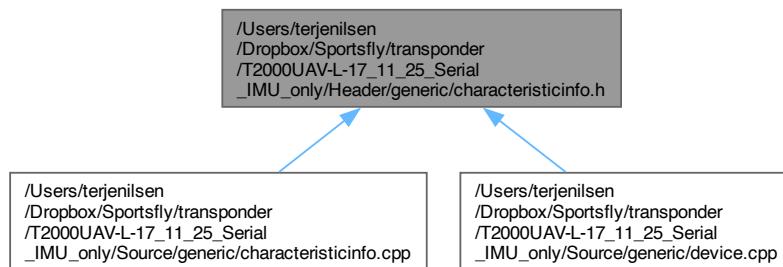
00001 #ifndef __WIT_C_SDK_H
00002 #define __WIT_C_SDK_H
00003
00004 #ifdef __cplusplus
00005 extern "C" {
00006 #endif
00007
00008 #include <stdint.h>
00009 #include <stdio.h>
00010 #include <string.h>
00011 #include "REG.h"
00012
00013 #define SLEEP_MS(ms) usleep((ms) * 1000)
00014 #define SCANF_INT(fmt, ptr) scanf(fmt, ptr)
00015
00016 // ---- types used by the SDK callbacks
00017 #ifdef Q_OS_ANDROID
00018 using UINT32 = uint32_t;
00019 #else
00020 #define UINT32 uint32_t
00021 #endif
00022
00023 #define WIT_HAL_OK      (0)
00024 #define WIT_HAL_BUSY    (-1)
00025 #define WIT_HAL_TIMEOUT (-2)
00026 #define WIT_HAL_ERROR   (-3)
00027 #define WIT_HAL_NOMEM   (-4)
00028 #define WIT_HAL_EMPTY   (-5)
00029 #define WIT_HAL_INVAL   (-6)
00030
00031 #define WIT_DATA_BUFF_SIZE 256
00032
00033 #define WIT_PROTOCOL_NORMAL      0
00034 #define WIT_PROTOCOL_MODBUS     1
00035 #define WIT_PROTOCOL_CAN        2
00036 #define WIT_PROTOCOL_I2C        3
00037 #define WIT_PROTOCOL_JY61       4
00038 #define WIT_PROTOCOL_905x_MODBUS 5
00039 #define WIT_PROTOCOL_905x_CAN   6
00040
00041 /* serial function */
00042 typedef void (*SerialWrite)(uint8_t *p_ucData, uint32_t uiLen);
00043 int32_t WitSerialWriteRegister(SerialWrite write_func);
00044
00045 /* Delayms function */
00046 typedef void (*DelaymsCb)(uint16_t ucMs);
00047 int32_t WitDelayMsRegister(DelaymsCb delayms_func);
00048
00049 typedef void (*RegUpdateCb)(uint32_t uiReg, uint32_t uiRegNum);
00050 int32_t WitRegisterCallBack(RegUpdateCb update_func);
00051 int32_t WitWriteReg(uint32_t uiReg, uint16_t usData);
00052 int32_t WitReadReg(uint32_t uiReg, uint32_t uiReadNum);
00053 int32_t WitInit(uint32_t uiProtocol, uint8_t ucAddr);
00054 void WitDeInit(void);
00055 void WitSerialDataIn(void* handler, const char* data, uint32_t length); // uint8_t ucData);
00056 void posix_delay_ms(uint16_t ms);
00057
00072 int32_t WitStartAccCali(void);
00073 int32_t WitStopAccCali(void);
00074 int32_t WitStartMagCali(void);
00075 int32_t WitStopMagCali(void);
00076 int32_t WitSetUartBaud(int32_t uiBaudIndex);
00077 int32_t WitSetBandwidth(int32_t uiBaudWidth);
00078 int32_t WitSetOutputRate(int32_t uiRate);
00079 int32_t WitSetContent(int32_t uiRsw);
00080 int32_t WitSetCanBaud(int32_t uiBaudIndex);
00081 int32_t WitSaveParameter(void);
00082 int32_t WitSetForReset(void);
00083 int32_t WitCaliRefAngle(void);
00084
00085 char CheckRange(short sTemp, short sMin, short sMax);
00086
00087 extern int16_t sReg[REGSIZE];
00088
00089 #ifdef __cplusplus
00090 }
00091 #endif
00092
00093 #endif /* __WIT_C_SDK_H */
```

9.47 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header/generic/characteristicinfo.h File Reference

```
#include <QObject>
#include <QString>
#include <QtBluetooth/QLowEnergyCharacteristic>
Include dependency graph for characteristicinfo.h:
```



This graph shows which files directly or indirectly include this file:



Data Structures

- class [CharacteristicInfo](#)

9.48 characteristicinfo.h

[Go to the documentation of this file.](#)

```
00001 /*****
00002 /**
00003 ** Copyright (C) 2013 BlackBerry Limited. All rights reserved.
00004 ** Copyright (C) 2017 The Qt Company Ltd.
00005 ** Contact: https://www.qt.io/licensing/
```

```

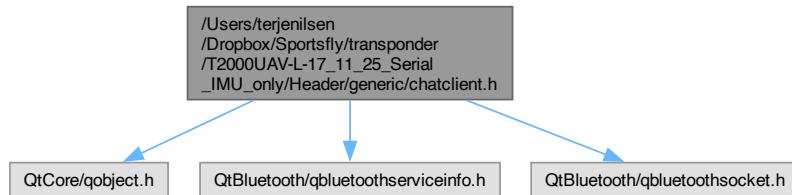
00006 /**
00007 ** This file is part of the QtBluetooth module of the Qt Toolkit.
00008 **
00009 ** $QT_BEGIN_LICENSE:BSD$
00010 ** Commercial License Usage
00011 ** Licensees holding valid commercial Qt licenses may use this file in
00012 ** accordance with the commercial license agreement provided with the
00013 ** Software or, alternatively, in accordance with the terms contained in
00014 ** a written agreement between you and The Qt Company. For licensing terms
00015 ** and conditions see https://www.qt.io/terms-conditions. For further
00016 ** information use the contact form at https://www.qt.io/contact-us.
00017 **
00018 ** BSD License Usage
00019 ** Alternatively, you may use this file under the terms of the BSD license
00020 ** as follows:
00021 **
00022 ** "Redistribution and use in source and binary forms, with or without
00023 ** modification, are permitted provided that the following conditions are
00024 ** met:
00025 **      * Redistributions of source code must retain the above copyright
00026 **            notice, this list of conditions and the following disclaimer.
00027 **      * Redistributions in binary form must reproduce the above copyright
00028 **            notice, this list of conditions and the following disclaimer in
00029 **            the documentation and/or other materials provided with the
00030 **            distribution.
00031 **      * Neither the name of The Qt Company Ltd nor the names of its
00032 **            contributors may be used to endorse or promote products derived
00033 **            from this software without specific prior written permission.
00034 **
00035 **
00036 ** THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
00037 ** "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
00038 ** LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
00039 ** A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
00040 ** OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
00041 ** SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
00042 ** LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
00043 ** DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
00044 ** THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
00045 ** (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
00046 ** OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE."
00047 **
00048 ** $QT_END_LICENSE$
00049 **
00050 *****/
00051
00052 #ifndef CHARACTERISTICINFO_H
00053 #define CHARACTERISTICINFO_H
00054 #include <QObject>
00055 #include <QString>
00056 #include <QtBluetooth/QLowEnergyCharacteristic>
00057
00058 class CharacteristicInfo: public QObject
00059 {
00060     Q_OBJECT
00061
00062 public:
00063     CharacteristicInfo() = default;
00064     CharacteristicInfo(const QLowEnergyCharacteristic &characteristic);
00065     void setCharacteristic(const QLowEnergyCharacteristic &characteristic);
00066     QString getName() const;
00067     QString getUuid() const;
00068     QString getValue() const;
00069     QString getPermission() const;
00070     QLowEnergyCharacteristic getCharacteristic() const;
00071
00072 Q_SIGNALS:
00073     void characteristicChanged();
00074
00075 private:
00076     QLowEnergyCharacteristic m_characteristic;
00077 };
00078
00079 #endif // CHARACTERISTICINFO_H

```

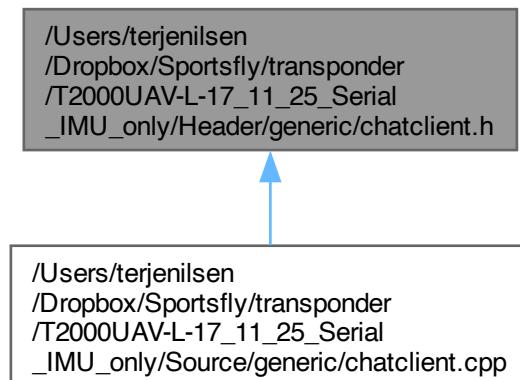
9.49 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_← 11_25_Serial_IMU_only/Header/generic/chatclient.h File Reference

```
#include <QtCore/qobject.h>
#include <QtBluetooth/qbluetoothserviceinfo.h>
```

```
#include <QtBluetooth/qbluetoothsocket.h>
Include dependency graph for chatclient.h:
```



This graph shows which files directly or indirectly include this file:



Data Structures

- class [ChatClient](#)
[declaration]

9.50 chatclient.h

[Go to the documentation of this file.](#)

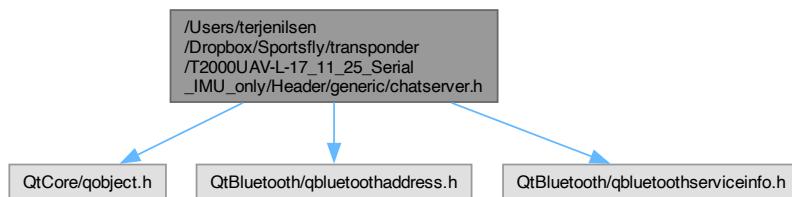
```

00001 // Copyright (C) 2017 The Qt Company Ltd.
00002 // SPDX-License-Identifier: LicenseRef-Qt-Commercial OR BSD-3-Clause
00003
00004 #ifndef CHATCLIENT_H
00005 #define CHATCLIENT_H
00006
00007 #include <QtCore/qobject.h>
00008
00009 #include <QtBluetooth/qbluetoothserviceinfo.h>
00010 #include <QtBluetooth/qbluetoothsocket.h>
00011
  
```

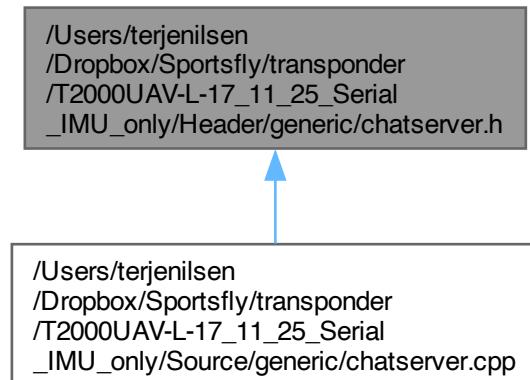
```
00012 QT_FORWARD_DECLARE_CLASS(QBluetoothSocket)
00013
00014 QT_USE_NAMESPACE
00015
00017 class ChatClient : public QObject
00018 {
00019     Q_OBJECT
00020
00021 public:
00022     explicit ChatClient(QObject *parent = nullptr);
00023     ~ChatClient();
00024
00025     void startClient(const QBluetoothServiceInfo &remoteService);
00026     void stopClient();
00027
00028 public slots:
00029     void sendMessage(const QString &message);
00030
00031 signals:
00032     void messageReceived(const QString &sender, const QString &message);
00033     void connected(const QString &name);
00034     void disconnected();
00035     void socketErrorOccurred(const QString &errorString);
00036
00037 private slots:
00038     void readSocket();
00039     void connected();
00040     void onSocketErrorOccurred(QBluetoothSocket::SocketError);
00041
00042 private:
00043     QBluetoothSocket *socket = nullptr;
00044 };
00046
00047 #endif // CHATCLIENT_H
```

9.51 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_← 11_25_Serial_IMU_only/Header/generic/chatserver.h File Reference

```
#include <QtCore/qobject.h>
#include <QtBluetooth/qbluetoothaddress.h>
#include <QtBluetooth/qbluetoothserviceinfo.h>
Include dependency graph for chatserver.h:
```



This graph shows which files directly or indirectly include this file:



Data Structures

- class [ChatServer](#)

[declaration]

9.52 chatserver.h

[Go to the documentation of this file.](#)

```

00001 // Copyright (C) 2017 The Qt Company Ltd.
00002 // SPDX-License-Identifier: LicenseRef-Qt-Commercial OR BSD-3-Clause
00003
00004 #ifndef CHATSERVER_H
00005 #define CHATSERVER_H
00006
00007 #include <QtCore/qobject.h>
00008
00009 #include <QtBluetooth/qbluetoothaddress.h>
00010 #include <QtBluetooth/qbluetoothserviceinfo.h>
00011
00012 QT_FORWARD_DECLARE_CLASS(QBluetoothServer)
00013 QT_FORWARD_DECLARE_CLASS(QBluetoothSocket)
00014
00015 QT_USE_NAMESPACE
00016
00017 class ChatServer : public QObject
00018 {
00019     Q_OBJECT
00020
00021     public:
00022         explicit ChatServer(QObject *parent = nullptr);
00023         ~ChatServer();
00024
00025         void startServer(const QBluetoothAddress &localAdapter = QBluetoothAddress());
00026         void stopServer();
00027
00028     public slots:
00029         void sendMessage(const QString &message);
00030
00031     signals:
00032         void messageReceived(const QString &sender, const QString &message);
00033         void clientConnected(const QString &name);
00034         void clientDisconnected(const QString &name);
00035
00036

```

```

00037 private slots:
00038     void clientConnected();
00039     void clientDisconnected();
00040     void readSocket();
00041
00042 private:
00043     QBluetoothServer *rfcommServer = nullptr;
00044     QBluetoothServiceInfo serviceInfo;
00045     QList<QBluetoothSocket *> clientSockets;
00046     QMap<QBluetoothSocket *, QString> clientNames;
00047 };
00049
00050 #endif // CHATSERVER_H

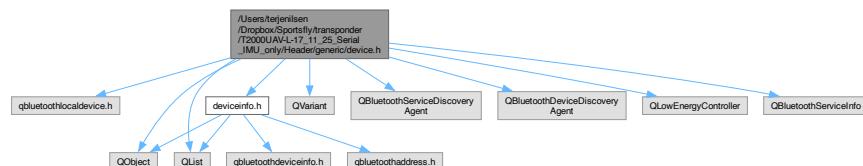
```

9.53 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_← 11_25_Serial_IMU_only/Header/generic/device.h File Reference

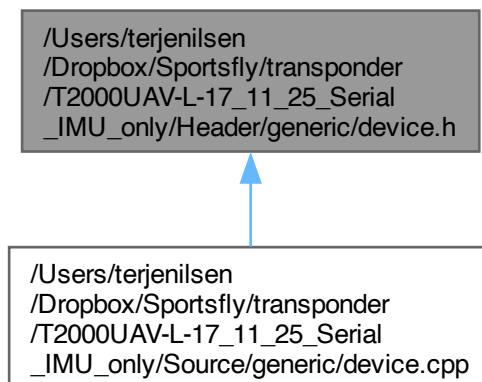
```

#include <qbluetoothlocaldevice.h>
#include <QObject>
#include <QVariant>
#include <QList>
#include <QBluetoothServiceDiscoveryAgent>
#include <QBluetoothDeviceDiscoveryAgent>
#include <QLowEnergyController>
#include <QBluetoothServiceInfo>
#include "deviceinfo.h"
Include dependency graph for device.h:

```



This graph shows which files directly or indirectly include this file:



Data Structures

- class Device

9.54 device.h

[Go to the documentation of this file.](#)

```

00001 /*****
00002 /**
00003 ** Copyright (C) 2017 The Qt Company Ltd.
00004 ** Contact: https://www.qt.io/licensing/
00005 /**
00006 ** This file is part of the demonstration applications of the Qt Toolkit.
00007 /**
00008 ** $QT_BEGIN_LICENSE:BSD
00009 ** Commercial License Usage
00010 ** Licensees holding valid commercial Qt licenses may use this file in
00011 ** accordance with the commercial license agreement provided with the
00012 ** Software or, alternatively, in accordance with the terms contained in
00013 ** a written agreement between you and The Qt Company. For licensing terms
00014 ** and conditions see https://www.qt.io/terms-conditions. For further
00015 ** information use the contact form at https://www.qt.io/contact-us.
00016 /**
00017 ** BSD License Usage
00018 ** Alternatively, you may use this file under the terms of the BSD license
00019 ** as follows:
00020 /**
00021 ** Redistributions and use in source and binary forms, with or without
00022 ** modification, are permitted provided that the following conditions are
00023 ** met:
00024 **      * Redistributions of source code must retain the above copyright
00025 **        notice, this list of conditions and the following disclaimer.
00026 **      * Redistributions in binary form must reproduce the above copyright
00027 **        notice, this list of conditions and the following disclaimer in
00028 **        the documentation and/or other materials provided with the
00029 **        distribution.
00030 **      * Neither the name of The Qt Company Ltd nor the names of its
00031 **        contributors may be used to endorse or promote products derived
00032 **        from this software without specific prior written permission.
00033 /**
00034 /**
00035 ** THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
00036 ** "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
00037 ** LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
00038 ** A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
00039 ** OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
00040 ** SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
00041 ** LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
00042 ** DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
00043 ** THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
00044 ** (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
00045 ** OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE."
00046 /**
00047 ** $QT_END_LICENSE$
00048 /**
00049 ****
00050
00051 #ifndef DEVICE_H
00052 #define DEVICE_H
00053
00054 #include <qbluetoothlocaldevice.h>
00055 #include <QObject>
00056 #include <QVariant>
00057 #include <QList>
00058 #include <QBluetoothServiceDiscoveryAgent>
00059 #include <QBluetoothDeviceDiscoveryAgent>
00060 #include <QLowEnergyController>
00061 #include <QBluetoothServiceInfo>
00062
00063 #include "deviceinfo.h"
00064
00065 QT_FORWARD_DECLARE_CLASS (QBluetoothDeviceInfo)
00066 QT_FORWARD_DECLARE_CLASS (QBluetoothServiceInfo)
00067
00068 class Device: public QObject
00069 {
00070     Q_OBJECT
00071
00072 public:
00073     Device();

```

```

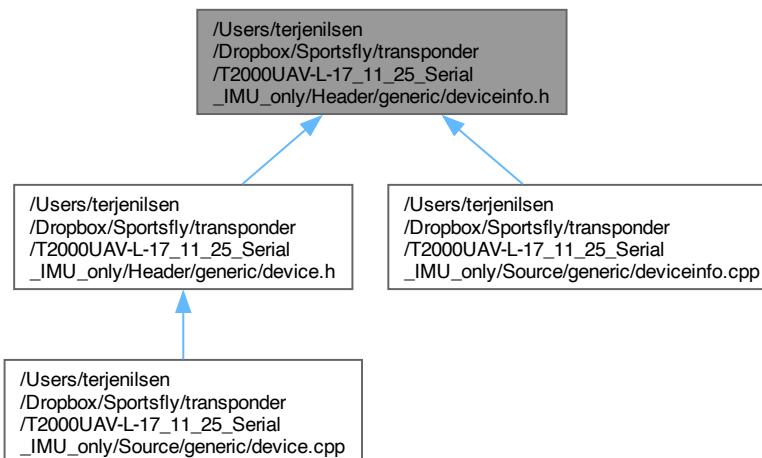
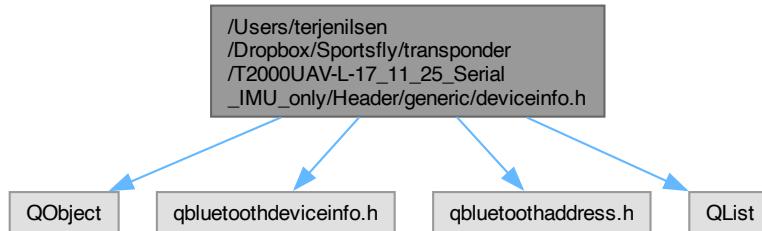
00074     ~Device();
00075     /*
00076     QVariant getDevices();
00077     QVariant getServices();
00078     QVariant getCharacteristics();
00079     QString getUpdate();
00080     */
00081     bool state();
00082     void sendData(QString data, int ch);
00083
00084     QString m_message;
00085     QLowEnergyService *l_service = nullptr;
00086     QString key2;
00087     QString receivedData;
00088
00089 public slots:
00090     void startDeviceDiscovery();
00091     void scanServices(const QString &address);
00092
00093     void connectToService(const QString &uuid);
00094     void disconnectFromDevice();
00095     void setVal();
00096     void test();
00097     void updateValueSet();
00098
00099 private slots:
00100     // QBluetoothDeviceDiscoveryAgent related
00101     void addDevice(const QBluetoothDeviceInfo&);
00102     void deviceScanFinished();
00103     void deviceScanError(QBluetoothDeviceDiscoveryAgent::Error);
00104
00105     // QLowEnergyController realted
00106     void addLowEnergyService(const QBluetoothUuid &uuid);
00107     void deviceConnected();
00108     void errorReceived(QLowEnergyController::Error);
00109     void serviceScanDone();
00110     void deviceDisconnected();
00111
00112     // QLowEnergyService related
00113     void serviceDetailsDiscovered(QLowEnergyService::ServiceState newState);
00114
00115 Q_SIGNALS:
00116     // void servicesUpdated();
00117     void characteristicsUpdated();
00118     void updateChanged();
00119     void stateChanged();
00120     void disconnected();
00121     void randomAddressChanged();
00122     void updateDisplay();
00123
00124 private:
00125     void setUpdate(const QString &message);
00126     QBluetoothDeviceDiscoveryAgent *discoveryAgent;
00127     DeviceInfo currentDevice;
00128     QList<QObject *> devices;
00129     QList<QObject *> m_services;
00130     QList<QObject *> m_characteristics;
00131     QString m_previousAddress;
00132     // QString m_message;
00133     bool connected = false;
00134     QLowEnergyController *controller = nullptr;
00135     bool m_deviceScanState = false;
00136     bool randomAddress = false;
00137
00138     void updateValue(const QLowEnergyCharacteristic &c, const QByteArray &value);
00139
00140
00141
00142
00143
00144 };
00145
00146 #endif // DEVICE_H

```

9.55 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_← 11_25_Serial_IMU_only/Header/generic/deviceinfo.h File Reference

```
#include <QObject>
#include <qbluetoothdeviceinfo.h>
```

```
#include <qbluetoothaddress.h>
#include <QList>
Include dependency graph for deviceinfo.h:
```



Data Structures

- class [DeviceInfo](#)

9.56 deviceinfo.h

[Go to the documentation of this file.](#)

```
00001 /*****
00002 /**
00003 ** Copyright (C) 2013 BlackBerry Limited. All rights reserved.
00004 ** Copyright (C) 2017 The Qt Company Ltd.
00005 ** Contact: https://www.qt.io/licensing/
00006 /**
00007 ** This file is part of the QtBluetooth module of the Qt Toolkit.
```

```

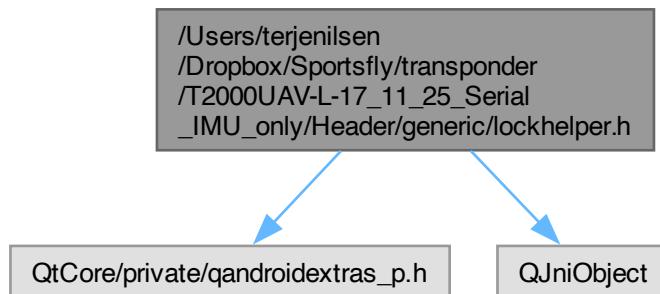
00008 /**
00009 ** $QT_BEGIN_LICENSE:BSD$
00010 ** Commercial License Usage
00011 ** Licensees holding valid commercial Qt licenses may use this file in
00012 ** accordance with the commercial license agreement provided with the
00013 ** Software or, alternatively, in accordance with the terms contained in
00014 ** a written agreement between you and The Qt Company. For licensing terms
00015 ** and conditions see https://www.qt.io/terms-conditions. For further
00016 ** information use the contact form at https://www.qt.io/contact-us.
00017 /**
00018 ** BSD License Usage
00019 ** Alternatively, you may use this file under the terms of the BSD license
00020 ** as follows:
00021 /**
00022 ** "Redistribution and use in source and binary forms, with or without
00023 ** modification, are permitted provided that the following conditions are
00024 ** met:
00025 ** * Redistributions of source code must retain the above copyright
00026 ** notice, this list of conditions and the following disclaimer.
00027 ** * Redistributions in binary form must reproduce the above copyright
00028 ** notice, this list of conditions and the following disclaimer in
00029 ** the documentation and/or other materials provided with the
00030 ** distribution.
00031 ** * Neither the name of The Qt Company Ltd nor the names of its
00032 ** contributors may be used to endorse or promote products derived
00033 ** from this software without specific prior written permission.
00034 /**
00035 /**
00036 ** THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
00037 ** "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
00038 ** LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
00039 ** A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
00040 ** OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
00041 ** SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
00042 ** LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
00043 ** DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
00044 ** THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
00045 ** (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
00046 ** OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE."
00047 /**
00048 ** $QT_END_LICENSE$
00049 /**
00050 *****/
00051
00052 #ifndef DEVICEINFO_H
00053 #define DEVICEINFO_H
00054
00055 #include <QObject>
00056 #include <qbluetoothdeviceinfo.h>
00057 #include <qbluetoothaddress.h>
00058 #include <QList>
00059
00060 class DeviceInfo: public QObject
00061 {
00062     Q_OBJECT
00063     Q_PROPERTY(QString deviceName READ getName NOTIFY deviceChanged)
00064
00065 public:
00066     DeviceInfo() = default;
00067     DeviceInfo(const QBluetoothDeviceInfo &d);
00068     QString getAddress() const;
00069     QString getName() const;
00070     QBluetoothDeviceInfo getDevice();
00071     void setDevice(const QBluetoothDeviceInfo &dev);
00072
00073 Q_SIGNALS:
00074     void deviceChanged();
00075
00076 private:
00077     QBluetoothDeviceInfo device;
00078 };
00079
00080 #endif // DEVICEINFO_H

```

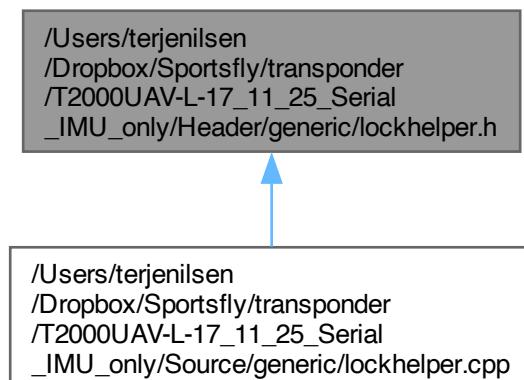
9.57 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_← 11_25_Serial_IMU_only/Header/generic/lockhelper.h File Reference

```
#include <QtCore/private/qandroidextras_p.h>
#include <QJniObject>
```

Include dependency graph for lockhelper.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- class [KeepAwakeHelper](#)

9.58 lockhelper.h

[Go to the documentation of this file.](#)

```

00001 #ifndef LOCKHELPER_H
00002 #define LOCKHELPER_H
00003
00004 #pragma once
00005
00006 #include <QtCore/private/qandroidextras_p.h>
  
```

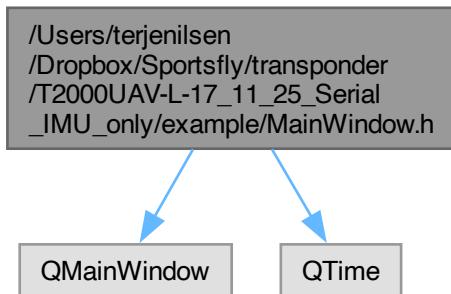
```

00007 #include <QJniObject>
00008
00009 class KeepAwakeHelper
00010 {
00011 public:
00012     KeepAwakeHelper();
00013     void EnableKeepAwakeHelper();
00014     virtual ~KeepAwakeHelper();
00015
00016 private:
00017     QJniObject m_wakeLock;
00018 };
00019
00020 #endif // LOCKHELPER_H

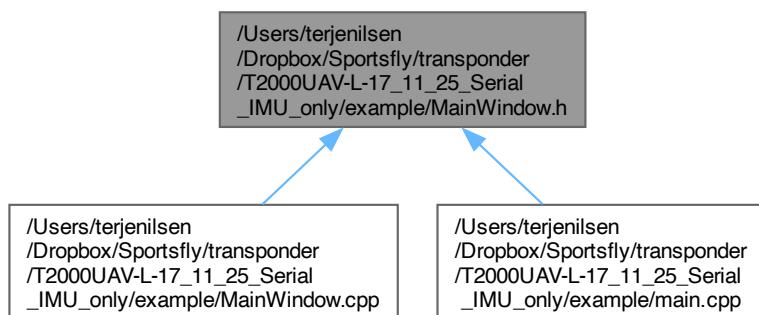
```

9.59 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/mainwindow.h File Reference

```
#include < QMainWindow>
#include < QTime>
Include dependency graph for MainWindow.h:
```



This graph shows which files directly or indirectly include this file:



Data Structures

- class [MainWindow](#)

Main UI window for the Glasscockpit 200-UAV.

Namespaces

- namespace [Ui](#)

9.60 MainWindow.h

[Go to the documentation of this file.](#)

```

00001 /***** Copyright (C) 2021 Marek M. Cel ****/*
00002 * Copyright (C) 2021 Marek M. Cel
00003 *
00004 * Permission is hereby granted, free of charge, to any person obtaining
00005 * a copy of this software and associated documentation files (the "Software"),
00006 * to deal in the Software without restriction, including without limitation
00007 * the rights to use, copy, modify, merge, publish, distribute, sublicense,
00008 * and/or sell copies of the Software, and to permit persons to whom
00009 * the Software is furnished to do so, subject to the following conditions:
00010 *
00011 * The above copyright notice and this permission notice shall be included
00012 * in all copies or substantial portions of the Software.
00013 *
00014 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS
00015 * OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
00016 * FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL
00017 * THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
00018 * LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
00019 * OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS
00020 * IN THE SOFTWARE.
00021 ****/
00022 #ifndef MAINWINDOW_H
00023 #define MAINWINDOW_H
00024
00025
00026
00027 #include <QMainWindow>
00028 #include <QTime>
00029
00030
00031
00032 namespace Ui
00033 {
00034     class MainWindow;
00035 }
00036
00037
00038
00039 class MainWindow : public QMainWindow
00040 {
00041     Q_OBJECT
00042
00043     public:
00044         explicit MainWindow( QWidget *parent = Q_NULLPTR );
00045
00046     protected:
00047         ~MainWindow();
00048
00049     private:
00050         void timerEvent( QTimerEvent *event );
00051
00052     private:
00053
00054     private:
00055
00056     private:
00057
00058     private:
00059
00060     private:
00061
00062     private:
00063
00064     private:
00065
00066     private:
00067
00068     private:
00069     private:
00070
00071     private:
00072     private:
00073 };
00074
00075
00076
00077 #endif // MAINWINDOW_H

```

9.61 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_← 11_25_Serial_IMU_only/Header/generic/mainwindow.h File Reference

```
#include <QMainWindow>
#include <QQuickView>
#include <QtCharts/QChartView>
#include <QtCharts/QSplineSeries>
#include <QEapsedTimer>
#include <QCameraDevice>
#include <QMediaDevices>
#include <QCamera>
#include <QMediaCaptureSession>
#include <QMediaRecorder>
#include <QImageCapture>
#include <QMessageBox>
#include <QSplashScreen>
#include <QTimer>
#include <QScreen>
#include <QSize>
#include <QDateTime>
#include <QPushButton>
#include <QPermission>
#include <QGyroscope>
#include <QGyroscopeReading>
#include <QAccelerometer>
#include <QAccelerometerReading>
#include <QCompass>
#include <QCompassReading>
#include <QMagnetometer>
#include <QMagnetometerReading>
#include <QOrientationSensor>
#include <QOrientationReading>
#include <QAmbientTemperatureSensor>
#include <QAmbientTemperatureReading>
#include <QRotationSensor>
#include <QPressureSensor>
#include <QPressureReading>
#include <QGeoPositionInfo>
#include <QGeoCoordinate>
#include <QGeoPositionInfoSource>
#include "mytcpsocket.h"
#include "ekfNavINS.h"
#include "mqttclient.h"
#include "example/WidgetSix.h"
```

Include dependency graph for mainwindow.h:



Data Structures

- class [MainWindow](#)

Main UI window for the Glasscockpit 200-UAV.

Namespaces

- namespace `Ui`

Macros

- `#define USE_KeepAwakeHelper`
- `#define IMAGES_DIR "/storage/emulated/0/DCIM/Camera"`
- `#define LOG_DIR "/storage/emulated/0/Documents"`
- `#define RADIO "/setup_radio_b.txt"`
- `#define AIRPLANE "/setup_ln_b.txt"`
- `#define CONFIG "/config_b.txt"`
- `#define FLIGHTLOG "/flightlog.txt"`
- `#define TRANSPONDERLOG "/log.txt"`
- `#define P_TRANSPOUNDER 0`
- `#define P_IMU 1`
- `#define P_FLIGT_INSTRUMENT 2`
- `#define P_GLASS_COCPIT 3`
- `#define P_RADAR 4`
- `#define P_RADIO_LIST 5`
- `#define P_AUTOPILOT 6`
- `#define P_CONFIG 7`
- `#define P_CAMERA 8`

Functions

- `void Qiskit (void)`

9.61.1 Macro Definition Documentation

9.61.1.1 AIRPLANE

```
#define AIRPLANE "/setup_ln_b.txt"
```

9.61.1.2 CONFIG

```
#define CONFIG "/config_b.txt"
```

9.61.1.3 FLIGHTLOG

```
#define FLIGHTLOG "/flightlog.txt"
```

9.61.1.4 IMAGES_DIR

```
#define IMAGES_DIR "/storage/emulated/0/DCIM/Camera"
```

9.61.1.5 LOG_DIR

```
#define LOG_DIR "/storage/emulated/0/Documents"
```

9.61.1.6 P_AUTOPILOT

```
#define P_AUTOPILOT 6
```

9.61.1.7 P_CAMERA

```
#define P_CAMERA 8
```

9.61.1.8 P_CONFIG

```
#define P_CONFIG 7
```

9.61.1.9 P_FLIGT_INSTRUMENT

```
#define P_FLIGT_INSTRUMENT 2
```

9.61.1.10 P_GLASS_COCPIT

```
#define P_GLASS_COCPIT 3
```

9.61.1.11 P_IMU

```
#define P_IMU 1
```

9.61.1.12 P_RADAR

```
#define P_RADAR 4
```

9.61.1.13 P_RADIO_LIST

```
#define P_RADIO_LIST 5
```

9.61.1.14 P_TRANSPOUNDER

```
#define P_TRANSPOUNDER 0
```

9.61.1.15 RADIO

```
#define RADIO "/setup_radio_b.txt"
```

9.61.1.16 TRANSPONDERLOG

```
#define TRANSPONDERLOG "/log.txt"
```

9.61.1.17 USE_KeepAwakeHelper

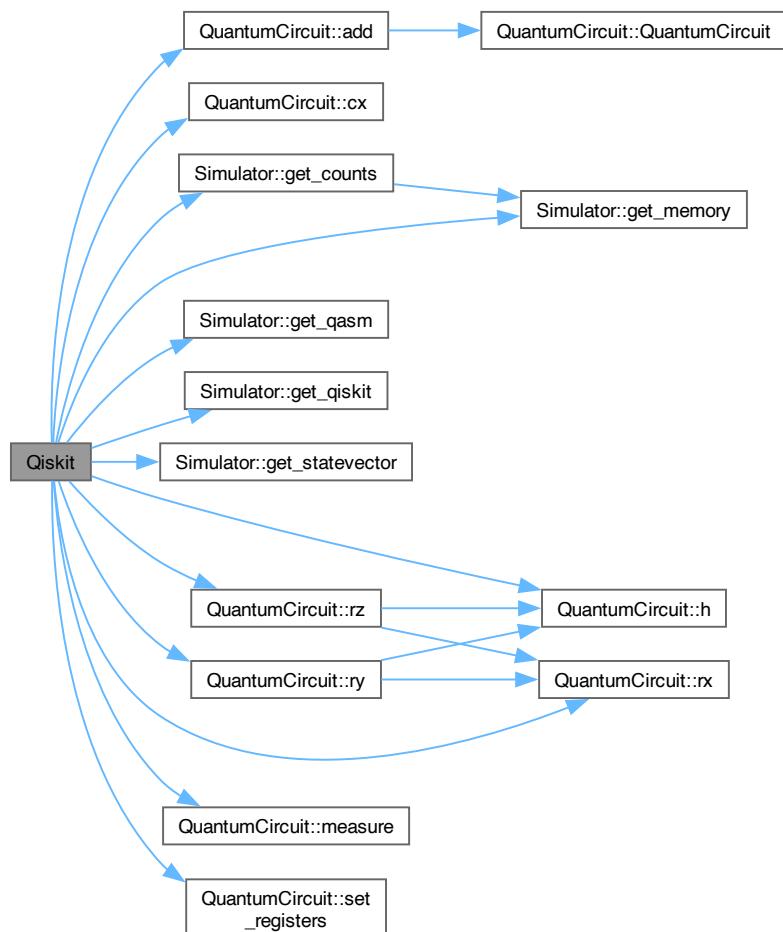
```
#define USE_KeepAwakeHelper
```

9.61.2 Function Documentation

9.61.2.1 Qiskit()

```
void Qiskit (
    void )
```

Here is the call graph for this function:



9.62 mainwindow.h

[Go to the documentation of this file.](#)

```
00001 #ifndef MAINWINDOW_H
00002 #define MAINWINDOW_H
00003
00004 #include <QMainWindow>
00005 #include <QQuickView>
00006
00007 #include <QtCharts/QChartView>
00008 #include <QtCharts/QSplineSeries>
00009
00010 #include <QEapsedTimer>
00011 #include <QCameraDevice>
00012 #include <QMediaDevices>
00013 #include <QCamera>
00014 #include <QMediaCaptureSession>
00015 #include <QMediaRecorder>
00016 #include <QImageCapture>
00017
00018 #include <QMessageBox>
00019 #include <QSplashScreen>
00020 #include <QTimer>
00021 #include <QScreen>
00022 #include <QSize>
00023 #include <QDateTime>
00024 #include <QPushButton>
00025 #include <QPermission>
00026
00027 #include <QGyroscope>
00028 #include <QGyroscopeReading>
00029
00030 #include <QAccelerometer>
00031 #include <QAccelerometerReading>
00032
00033 #include <QCompass>
00034 #include <QCompassReading>
00035
00036 #include <QMagnetometer>
00037 #include <QMagnetometerReading>
00038
00039 #include <QOrientationSensor>
00040 #include <QOrientationReading>
00041
00042 #include <QAmbientTemperatureSensor>
00043 #include <QAmbientTemperatureReading>
00044
00045 #include <QRotationSensor>
00046 #include <QPressureSensor>
00047 #include <QPressureReading>
00048
00049 #include <QGeoPositionInfo>
00050 #include <QGeoCoordinate>
00051 #include <QGeoPositionInfoSource>
00052
00053 #include "mytcpsocket.h"
00054 #include "ekfNavINS.h"
00055 #include "mqttclient.h"
00056
00057 // Flight instruments
00058 #include "example/WidgetSix.h"
00059 // (WidgetSix includes WidgetAI/ALT/ASI/HI/TC/VSI/EADI/EHSI etc.)
00060
00061 #ifdef Q_OS_IOS
00062 #undef Q_OS_MAC
00063 #endif
00064
00065 QT_BEGIN_NAMESPACE
00066 namespace Ui { class SCREEN; }
00067 QT_END_NAMESPACE
00068
00069 // Enable Android keep-alive helper (no-op on other platforms)
00070 #define USE_KeepAwakeHelper
00071
00072 // -----
00073 // Platform-specific log/image directories
00074 // -----
00075 #ifdef Q_OS_IOS
00076 // iOS: user-visible Documents directory
00077 #define LOG_DIR    QStandardPaths::writableLocation(QStandardPaths::DocumentsLocation)
00078 #define IMAGES_DIR QStandardPaths::writableLocation(QStandardPaths::DocumentsLocation)
00079 #elif defined(Q_OS_MAC)
00080 // macOS: also use Documents
00081 #define IMAGES_DIR QStandardPaths::writableLocation(QStandardPaths::DocumentsLocation)
00082 #define LOG_DIR     QStandardPaths::writableLocation(QStandardPaths::DocumentsLocation)
```

```
00083 #else
00084 // Android: explicit external storage paths
00085 #define IMAGES_DIR "/storage/emulated/0/DCIM/Camera"
00086 #define LOG_DIR     "/storage/emulated/0/Documents"
00087 #endif
00088
00089 // -----
00090 // File names (relative to LOG_DIR)
00091 // -----
00092 #define RADIO          "/setup_radio_b.txt"
00093 #define AIRPLANE        "/setup_ln_b.txt"
00094 #define CONFIG          "/config_b.txt"
00095 #define FLIGHTLOG        "/flightlog.txt"
00096 #define TRANSPONDERLOG   "/log.txt"
00097
00098 // Define the Qiskit interface (used for performance test)
00099 void Qiskit(void);
00100
00113 class MainWindow : public QMainWindow
00114 {
00115     Q_OBJECT
00116
00117 public:
00118     explicit MainWindow(QWidget *parent = nullptr);
00119     ~MainWindow();
00120
00121 // -----
00122 // High-level configuration / persistence
00123 // -----
00124
00128     void addnext(int x);
00129
00133     void addcurrent(int x);
00134
00144     void setmode(int mode);
00145
00150     int set_default_radio(void);
00151
00156     int set_default_planes(void);
00157
00163     int set_default_config(const Matrix3x6 &sensor);
00164
00170     int get_default_config(Matrix3x6 &sensor);
00171
00175     void logLanded();
00176
00180     void logTakeoff();
00181
00185     void init();
00186
00190     void updateCameras();
00191
00195     void hideCamera();
00196
00200     void setCamera(const QCameraDevice &cameraDevice);
00201
00205     void permissionUpdated(const QPermission &permission);
00206
00210     void setButtonIcon(QString iconPath, QPushButton *button);
00211
00216     double getBearing(double lat1, double lon1, double lat2, double lon2);
00217
00221     void AccelerometerRead();
00222
00227     double setQNH();
00228
00232     void showImage();
00233
00235     QSplashScreen *splash = nullptr;
00236
00237 // -----
00238 // Attitude / navigation state (public so instruments can be read easily)
00239 // -----
00240
00242     Vector3d m_attitude;
00243
00245     double roll_blended = 0.0;
00246     bool    roll_blended_ok = false;
00247
00249     Matrix3x3 rotationMatrix;
00250
00252     Qt::ScreenOrientation ScreenMode;
00253
00255     int next[4]    = {7, 0, 0, 0};
00256
00258     int current[4] = {8, 8, 8, 8};
00259
```

```
00261     int mode = 0;
00262
00264     Vector3d m_install;
00265
00266     double m_vario = 0.0;
00267
00268     double takeoff_latitude = 0.0;
00269     double takeoff_longitude = 0.0;
00270     double takeoff_altitude = 0.0;
00271
00272     double m_tansALT = 0.0;
00273
00274     bool m_use_imu = false;
00275
00276 //    double m_speed = 0.0;
00277
00278     double m_head = 9999.0;
00279
00280     qreal m_temp = 9999.0;
00281
00282     double m_roll_angle = 0.0;
00283
00284     double m_total_accel = 0.0;
00285
00286     double m_var_speed = 0.0;
00287
00288     double m_ms = 0.0;
00289
00290     int Radar_depth = 120;
00291     int Radar_Height = 120;
00292
00293
00294
00295
00296
00297
00298
00299
00300
00301
00302
00303
00304
00305
00306
00307     bool m_bluetoothrunning = false;
00308
00309
00310     QDateTime m_takeoffTime;
00311     QDateTime m_landedTime;
00312
00313
00314     MyTcpSocket *mysocket = nullptr;
00315
00316 // -----
00317 // Page indices in stacked widget
00318 // -----
00319 #define P_TRANSPONDER 0
00320 #define P_IMU 1
00321 #define P_FLIGT_INSTRUMENT 2
00322 #define P_GLASS_COCKPIT 3
00323 #define P_RADAR 4
00324 #define P_RADIO_LIST 5
00325 #define P_AUTOPILOT 6
00326 #define P_CONFIG 7
00327 #define P_CAMERA 8
00328
00329     int currentIndex = 0;
00330
00331
00332 // -----
00333 // Default USB serial numbers / IDs (SIM vs REAL)
00334 // -----
00335     QString _transponder_id = "4150323833373009";
00336 //    QString _transponder_id = "4150323833373205"; ///< REAL
00337
00338     QString _radar_id = "415032383337320B";
00339 //    QString _radar_id = "4150325537323317"; ///< REAL
00340
00341 //    QString _IMU_id = "4150323833373009"; //< Default IMU ID (SIM).
00342 //    QString _IMU_id = "FTM2H8X"; //< REAL
00343     QString _IMU_id = "4150323833373205";
00344     static void setIMU(void *parent, bool use_imu);
00345
00346     void onResized(int);
00347
00348     QScreen *getActiveScreen(QWidget *pWidget) const;
00349
00350     NoButtonMessageBox *m_msgBox = nullptr;
00351
00352
00353 // -----
00354 // QML / Charts / Timing
00355 // -----
00356     QQuickView view;
00357     QSplineSeries *series = nullptr;
00358     QEapsedTimer m_timer;
00359
00360
00361 // -----
00362 // Instrument widgets (Qt analog style)
00363 // -----
00364     WidgetAI *_widgetAI = nullptr;
00365     WidgetTC *_widgetTC = nullptr;
00366     WidgetALT *_widgetALT = nullptr;
00367     WidgetASI *_widgetASI = nullptr;
```

```

00379     WidgetVSI * _widgetVSI = nullptr;
00380     WidgetHI * _widgetHI = nullptr;
00381     WidgetEADI * _widgetEADI = nullptr;
00382     WidgetEHSI * _widgetEHSI = nullptr;
00383
00384     // Convenience getters
00385     WidgetAI *getAI () { return _widgetAI; }
00386     WidgetTC *getTC () { return _widgetTC; }
00387     WidgetALT *getALT () { return _widgetALT; }
00388     WidgetASI *getASI () { return _widgetASI; }
00389     WidgetVSI *getVSI () { return _widgetVSI; }
00390     WidgetHI *getHI () { return _widgetHI; }
00391     WidgetEADI *getEADI () { return _widgetEADI; }
00392     WidgetEHSI *getEHJSI () { return _widgetEHSI; }
00393
00394 private:
00395     // -----
00396     // Camera / media
00397     // -----
00398     QActionGroup *videoDevicesGroup = nullptr;
00399     QMediaDevices m_devices;
00400     QScopedPointer<QCamera> m_camera;
00401     QMediaCaptureSession m_captureSession;
00402     QMediaRecorder *m_recorder = nullptr;
00403     QCameraDevice *m_cameraDevice = nullptr;
00404     QImageCapture *m_capture = nullptr;
00405
00406     // -----
00407     // Timers
00408     // -----
00409     bool alt_receiced = false;
00410     QTimer *m_Clock = nullptr;
00411     QTimer *timerAlt = nullptr;
00412     QTimer *timerPing = nullptr;
00413     QTimer *timerActive = nullptr;
00414     QTimer *timerpaint = nullptr;
00415     QTimer *timertakePicture= nullptr;
00416     QTimer *m_IMU = nullptr;
00417     QTimer *m_Display = nullptr;
00418
00419     // -----
00420     // EKF and attitude filter
00421     // -----
00422     ekfNavINS ekf;
00423
00424     int alt_mode = 1;
00425     QSize *m_size = nullptr;
00426
00427     void setalt(int alt_mode);
00428
00429     int m_reading = 0;
00430     int m_first = 0;
00431     int m_calibrate = 150;
00432     bool m_armed = false;
00433     bool m_takeoff = false;
00434     double m_bearing = 999.0;
00435     double m_heading = 0.0;
00436
00437     bool m_use_gps_in_attitude = false;
00438
00439     // -----
00440     // Sensors (QtSensors / QtPositioning)
00441     // -----
00442     // QAltimeterSensor *m_altimeter_sensor; // reserved
00443
00444     QPressureSensor *m_pressure_sensor = nullptr;
00445     QPressureReading *m_pressure_reader = nullptr;
00446
00447     QOrientationSensor *m_orientation_sensor = nullptr;
00448     QOrientationReading *m_orientation_reader = nullptr;
00449
00450     QRotationSensor *m_rotation_sensor = nullptr;
00451     QRotationReading *m_rotation_reader = nullptr;
00452
00453     QCompass *m_compass_sensor = nullptr;
00454     QCompassReading *m_compass_reader = nullptr;
00455
00456     QAccelerometer *m_accel_sensor = nullptr;
00457     QAccelerometerReading *m_accel_reader = nullptr;
00458     double m_acc_Y_calib = 0.0;
00459
00460     QGyroscope *m_gyro_sensor = nullptr;
00461     QGyroscopeReading *m_gyro_reader = nullptr;
00462
00463     QMagnetometer *m_mag_sensor = nullptr;
00464     QMagnetometerReading *m_mag_reader = nullptr;
00465

```

```
00466     QAmbientTemperatureSensor *m_temp_sensor = nullptr;
00467     QAmbientTemperatureReading *m_temp_reader = nullptr;
00468
00469     QGeoPositionInfoSource *m_geoPositionInfo = nullptr;
00470
00471 // -----
00472 // Misc state used in EKF / attitude blend
00473 // -----
00474     qreal m_offset      = 0.0;
00475     double heading_offset = 0.0;
00476     bool m_geopos       = false;
00477     double m_head_dir   = 0.0;
00478     double m_dt          = 0.0;
00479     Vector3d m_accel_body;
00480
00481     double a_pitch        = 0.0;
00482     double a_roll         = 0.0;
00483     double a_yaw          = 0.0;
00484     double m_pitch        = 0.0;
00485     double m_roll         = 0.0;
00486     double m_yaw          = 0.0;
00487
00488
00489     NoButtonMessageBox *m_msgBoxCalibrating = nullptr;
00490
00491 #if defined(Q_OS_ANDROID) && defined(USE_KeepAwakeHelper)
00492     KeepAwakeHelper *helper = new KeepAwakeHelper();
00493 #endif
00494
00495     void accepted();
00496
00500     void calcPosition(double vel_D);
00501
00502     void afterPressureReadingChanged();
00503
00504
00505 private slots:
00506 // -----
00507 // Simple UI slots / transponder keypad / exit etc.
00508 // -----
00509     void setVal();
00510
00511     // Keypad buttons (set "next" code digit)
00512     void on_pushButton_clicked();
00513     void on_pushButton_2_clicked();
00514     void on_pushButton_19_clicked();
00515     void on_pushButton_3_clicked();
00516     void on_pushButton_4_clicked();
00517     void on_pushButton_5_clicked();
00518     void on_pushButton_6_clicked();
00519     void on_pushButton_7_clicked();
00520     void on_pushButton_8_clicked();
00521     void on_pushButton_9_clicked();
00522     void on_pushButton_17_clicked();
00523     void on_pushButton_16_clicked();
00524     void on_pushButton_18_clicked();
00525
00526     // Transponder control
00527     void on_pushButton_Ident_clicked();
00528     void on_pushButton_stby_clicked();
00529     void on_pushButton_off_clicked();
00530     void on_pushButton_norm_clicked();
00531     void on_pushButton_alt_clicked();
00532     void on_pushButton_12_clicked();
00533     void on_pushButton_13_clicked();
00534
00535     // Misc controls
00536     void on_reconnect_now_clicked();
00537     void on_pushButton_20_clicked();
00538     void on_reset_heading_clicked();
00539     void on_select_transponder_page_3_clicked();
00540     void on_select_from_4_to_5_clicked();
00541
00542     // Status / timers
00543     void doCheck();
00544     void reset_ping();
00545     void active_ping();
00546     void doClock();
00547
00548     // GPS / position updates
00549     void positionUpdated(QGeoPositionInfo geoPositionInfo);
00550
00551     // Sensor change notifications
00552     void onRotationReadingChanged();
00553     void onReadingChanged();
00554     void onPressureReadingChanged();
00555     void onCompassReadingChanged();
```

```

00556     void onAccelerometerReadingChanged();
00557     void onGyroReadingChanged();
00558     void onMagReadingChanged();
00559     void onOrientationReadingChanged();
00560     void onTempReadingChanged();
00561
00562     // Page navigation (stackedWidget)
00563     void on_select_gyro_page_clicked();
00564     void on_select_gyro_page2_clicked();
00565     void on_select_camera_from_transponder_clicked();
00566     void on_select_dumy_page2_clicked();
00567     void on_select_transponder_page_clicked();
00568     void on_select_transponder_page2_clicked();
00569     void on_imu_reset_clicked();
00570     void on_timer_start_clicked();
00571     void onTextEdit_1TextChanged();
00572     void onTextEdit_2TextChanged();
00573     void onPushButton_15_clicked();
00574     void on_select_transponder_page2_2_clicked();
00575     void on_select_gyro_page2_2_clicked();
00576     void on_select_transponder_page2_3_clicked();
00577     void on_select_gyro_page2_3_clicked();
00578     void onPushButton_21_clicked();
00579     void onPushButton_22_clicked();
00580     void on_select_dumy_page2_2_clicked();
00581     void on_select_transponder_page_2_clicked();
00582     void on_select_page2_map_clicked();
00583     void on_select_transponder_page_camera_clicked();
00584     void on_use_gps_in_attitude_clicked();
00585     void on_select_transponder_page_4_clicked();
00586     void on_select_from_5_to_6_clicked();
00587     void on_use_built_inn_barometer_clicked();
00588
00589     void onPushButton_23_clicked();
00590     void on_exit_2_clicked();
00591     void on_use_hw_clicked();
00592     void on_reset_att_clicked();
00593     void on_fly_home_clicked();
00594
00595     void on_reset_altitude_3_clicked();
00596     void on_reset_altitude_2_clicked();
00597     void on_reset_heading_2_clicked();
00598     void on_use_ins_only_clicked();
00599
00600     // Camera
00601     void takePicture();
00602     void on_dial_valueChanged(int value);
00603     void on_dial_2_valueChanged(int value);
00604
00605     // EKF main loop
00606     void EKF();
00607
00615     static void getVal(void *parent, const char *data, uint32_t lenght);
00616
00617 public:
00619     int screen_index = 0;
00620
00623     Ui::SCREEN *ui = (Ui::SCREEN *) &(*new (Ui::SCREEN));
00624 }
00625
00626 #endif // MAINWINDOW_H

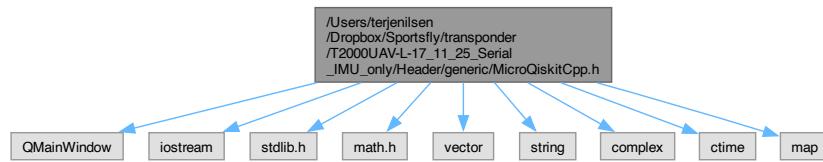
```

9.63 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_← 11_25_Serial_IMU_only/Header/generic/MicroQiskitCpp.h File Reference

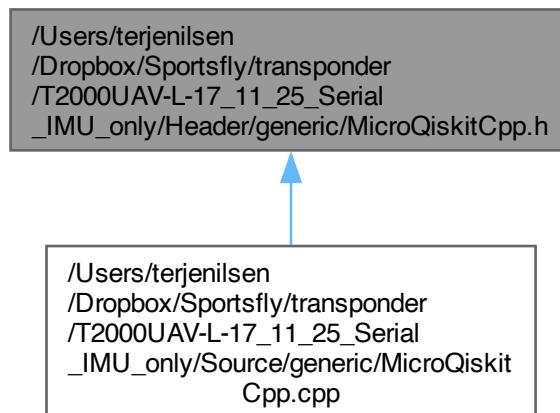
```
#include <QMainWindow>
#include <iostream>
#include <stdlib.h>
#include <math.h>
#include <vector>
#include <string>
#include <complex>
#include <ctime>
```

```
#include <map>
```

Include dependency graph for MicroQiskitCpp.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- class [QuantumCircuit](#)
- class [Simulator](#)

Macros

- #define [Qt](#)
- #define [RESET](#) "\033[0m"
- #define [RED](#) "\033[31m" /* Red */
- #define [ERROR](#)(MESSAGE)

Functions

- void [error_handler](#) (const string message)

9.63.1 Macro Definition Documentation

9.63.1.1 ERROR

```
#define ERROR(
    MESSAGE)
```

Value:

`error_handler(MESSAGE)`

9.63.1.2 Qt

```
#define Qt
```

9.63.1.3 RED

```
#define RED "\033[31m" /* Red */
```

9.63.1.4 RESET

```
#define RESET "\033[0m"
```

9.63.2 Function Documentation

9.63.2.1 error_handler()

```
void error_handler (
    const string message)
```

9.64 MicroQiskitCpp.h

[Go to the documentation of this file.](#)

```
00001 /*
00002 Comments in this file are mostly there to point out differences with the Python version of
00003 MicroQiskit.
00004 Access the lastest official release here:
00005     https://github.com/qiskit-community/MicroQiskit/tree/master/versions/C%2B%2B
00006 For comments on what everything is supposed to do, see the Python version of MicroQiskit.
00007     https://github.com/quantumjim/MicroQiskit/blob/master/microqiskit.py
00008     https://github.com/qiskit-community/MicroQiskit
00009 The present version of MicroQiskit in C++ has been the work of Omar Costa Hamdo and Dr. James Wootton
00010
00011 In development on repl.it (@quantum_jim/MicroQiskitC) and on github
00012     (https://github.com/omarcostahamido/MicroQiskitCpp).
00013 */
00014 #ifndef MICROQISKITCPP_H
00015 #define MICROQISKITCPP_H
00016 #include <QMainWindow>
00017 #include <iostream>
00018 #include <stdlib.h>
00019 #include <math.h>
```

```

00020 #include <vector>
00021 #include <string>
00022 #include <complex>
00023 #include <ctime>
00024 #include <map>
00025
00026 //##include <QtCore/QLoggingCategory>
00027 //##define qDebug() cout
00028 #define Qt
00029
00030 #define RESET "\033[0m"
00031 #define RED "\033[31m" /* Red */
00032 #define ERROR(MESSAGE) error_handler(MESSAGE)
00033
00034 using namespace std;
00035
00036 void error_handler(const string message)
00037 {
00038     qDebug() << RED << message << RESET << Qt::endl;
00039     abort();
00040 }
00041
00042 class QuantumCircuit {
00043
00044 public:
00045
00046     int nQubits, nBits;
00047     vector<vector<string> > data;
00048
00049     QuantumCircuit () {
00050
00051     }
00052     QuantumCircuit (int n, int m = 0) {
00053         set_registers (n, m);
00054     }
00055
00056     void set_registers (int n, int m = 0) {
00057         nQubits = n;
00058         nBits = m;
00059         if (!(nQubits==nBits || nBits==0))
00060         {
00061             ERROR("Only the cases nQubits=nBits and nBits=0 are allowed in MicroQiskit");
00062         }
00063     }
00064
00065     void add (QuantumCircuit qc2) {
00066
00067         nBits = max(nBits,qc2.nBits);
00068         nQubits = max(nQubits,qc2.nQubits);
00069         for (int g=0; g<qc2.data.size(); g++) {
00070             data.push_back( qc2.data[g] );
00071         }
00072     }
00073
00074     void initialize (vector<double> p){
00075         vector<string> init;
00076         //verify if the size of double vector is correct
00077         int t = pow(2, nQubits);
00078         if( !(p.size()==t||p.size()==t*2) ){
00079             ERROR("initialize: Can't initialize circuit. Please insert a vector {} with either
"+to_string(t)+" or "+to_string(t*2)+" doubles");
00080         }
00081         data.clear();
00082         init.push_back("init");
00083         init.push_back(to_string(p.size()));
00084         for(int i=0;i<p.size();i++){
00085             init.push_back(to_string(p[i]));
00086         }
00087         data.push_back(init);
00088     }
00089     void x (int q) {
00090         vector<string> gate;
00091         verify_qubit_range(q,"x gate");
00092         gate.push_back('x');
00093         gate.push_back(to_string(q));
00094         data.push_back(gate);
00095     }
00096     void rx (double theta, int q) {
00097         vector<string> gate;
00098         verify_qubit_range(q,"rx gate");
00099         gate.push_back("rx");
00100         gate.push_back(to_string(theta));
00101         gate.push_back(to_string(q));
00102         data.push_back(gate);
00103     }
00104     void h (int q) {
00105         vector<string> gate;

```

```

00106     verify_qubit_range(q, "h gate");
00107     gate.push_back("h");
00108     gate.push_back(to_string(q));
00109     data.push_back(gate);
00110 }
00111 void cx (int s, int t) {
00112     vector<string> gate;
00113     verify_qubit_range(s, "cx gate");
00114     verify_qubit_range(t, "cx gate");
00115     gate.push_back("cx");
00116     gate.push_back(to_string(s));
00117     gate.push_back(to_string(t));
00118     data.push_back(gate);
00119 }
00120 //new ch gate
00121 void ch (int s, int t) {
00122     vector<string> gate;
00123     verify_qubit_range(s, "ch gate");
00124     verify_qubit_range(t, "ch gate");
00125     gate.push_back("ch");
00126     gate.push_back(to_string(s));
00127     gate.push_back(to_string(t));
00128     data.push_back(gate);
00129 }
00130 //new crx gate
00131 void crx (double theta, int s, int t) {
00132     vector<string> gate;
00133     verify_qubit_range(s, "crx gate");
00134     verify_qubit_range(t, "crx gate");
00135     gate.push_back("crx");
00136     gate.push_back(to_string(theta));
00137     gate.push_back(to_string(s));
00138     gate.push_back(to_string(t));
00139     data.push_back(gate);
00140 }
00141 void measure (int q, int b) {
00142     vector<string> gate;
00143     if (!(q==b) )
00144     {
00145         ERROR("It is only possible to add measure gates of the form measure(j,j) in MicroQiskit");
00146     }
00147     verify_qubit_range(q, "measure gate");
00148     verify_bit_range(b, "measure gate");
00149
00150     gate.push_back("m");
00151     gate.push_back(to_string(b));
00152     gate.push_back(to_string(q));
00153     data.push_back(gate);
00154 }
00155 void rz (double theta, int q) {
00156     verify_qubit_range(q, "rz gate");
00157     h(q);
00158     rx(theta,q);
00159     h(q);
00160 }
00161 void ry (double theta, int q) {
00162     verify_qubit_range(q, "ry gate");
00163     rx(M_PI/2,q);
00164     h(q);
00165     rx(theta,q);
00166     h(q);
00167     rx(-M_PI/2,q);
00168 }
00169 void z (int q) {
00170     verify_qubit_range(q, "z gate");
00171     rz(M_PI,q);
00172 }
00173 void y (int q) {
00174     verify_qubit_range(q, "y gate");
00175     z(q);
00176     x(q);
00177 }
00178
00179 bool has_measurements(){
00180     //this is not totally bulletproof. i.e. it doesn't care where in time you actually place the
00181     //gates :
00182     vector<int> mGates;
00183     map<int,int> mUnique;
00184     map<int,int>::iterator it;
00185     //check all gates in circuit
00186     for (int g=0; g<data.size(); g++){
00187         //collect all measure gates in mGates
00188         if (data[g][0]=="m"){
00189             //just need the qubit they are assigned to
00190             mGates.push_back( stoi(data[g][1]) );
00191         }
00191     }

```

```

00192     //a full set of measurement gates must have a measure gate on each qubit in the circuit
00193     //create a list of all unique measure-gated qubits
00194     for(int num : mGates) {
00195         mUnique[mGates[num]] = 1;
00196     }
00197     //check if we have a measure gate for each qubit
00198     for(int i=0; i<nQubits; i++) {
00199         int it = mUnique.find(i);
00200         if(it == mUnique.end()) {
00201             return false;
00202         }
00203     }
00204
00205     return true;
00206 }
00207
00208 private:
00209     void verify_qubit_range(int q, string gate){
00210         if(!(q>=0) || !(q<nQubits) )
00211         {
00212             ERROR(gate+": Index for qubit out of range");
00213         }
00214     }
00215
00216     void verify_bit_range(int b, string gate){
00217         if(!(b>=0) || !(b<nBits))
00218         {
00219             ERROR(gate+": Index for bit out of range");
00220         }
00221     }
00222 }
00223
00224 };
00225
00226 class Simulator {
00227     // Contains methods required to simulate a circuit and provide the desired outputs.
00228
00229     vector<vector<double> > simulate (QuantumCircuit qc) {
00230
00231         vector<vector<double> > ket;
00232
00233         // initializing the internal ket
00234         for (int j=0; j<pow(2,qc.nQubits); j++){
00235             vector<double> e;
00236             for (int k=0; k<2; k++){
00237                 e.push_back(0.0);
00238             }
00239             ket.push_back(e); //add vector{0.0, 0.0}
00240         } //e.g. for 2 qubits <0.0, 0.0> <0.0, 0.0> <0.0, 0.0> <0.0, 0.0>
00241         ket[0][0] = 1.0; //change the first number on the first vector in ket. this means that by
00242         //default it will be measuring 0, because that's the first bitstr.
00243         //e.g. <1.0, 0.0> <0.0, 0.0> <0.0, 0.0> <0.0, 0.0>
00244
00245         //for each gate in qc.data vector (a vector which is of the type vector<vector<string> >) there
00246         //is an added vector<string>. Thus, qc.data.size() = the number of gates in qc.
00247         for (int g=0; g<qc.data.size(); g++){
00248
00249             if ( (qc.data[g][0]=="init") ){
00250                 // initialize
00251                 int initsize = stoi(qc.data[g][1]);
00252                 for(int i=0; i<initsize; i++){
00253                     if(initsize==pow(2,qc.nQubits)){
00254                         //if just a simple list
00255                         ket[i][0] = stod(qc.data[g][2+i]);
00256                         ket[i][1] = 0.0;
00257                     } else {
00258                         //else it must be a complete list
00259                         ket[i/2][i%2] = stod(qc.data[g][2+i]);
00260                     }
00261                 }
00262             } else if ( (qc.data[g][0]=="x") or (qc.data[g][0]=="rx") or (qc.data[g][0]=="h") ) {
00263
00264                 int q;
00265                 q = stoi( qc.data[g][qc.data[g].size()-1] );
00266                 //retrieve the last qubit number from the gate vector (target) as an int, e.g.
00267                 //<"h", "0"> = 0
00268
00269                 for (int i0=0; i0<pow(2,q); i0++){
00270                     for (int i1=0; i1<pow(2,qc.nQubits-q-1); i1++){
00271                         int b0,b1;
00272                         b0 = i0 + int(pow(2,q+1)) * i1;
00273                         b1 = b0 + int(pow(2,q));
00274
00275                         vector<double> e0, e1;
00276                         e0 = ket[b0];
00277                         e1 = ket[b1];
00278
00279                     }
00280                 }
00281             }
00282         }
00283     }
00284 }
```

```

00276             if (qc.data[g][0]==="x") {
00277                 ket[b0] = e1;
00278                 ket[b1] = e0;
00279             } else if (qc.data[g][0]==="rx") {
00280                 double theta = stof( qc.data[g][1] );
00281                 ket[b0][0] = e0[0]*cos(theta/2)+e1[1]*sin(theta/2);
00282                 ket[b0][1] = e0[1]*cos(theta/2)-e1[0]*sin(theta/2);
00283                 ket[b1][0] = e1[0]*cos(theta/2)+e0[1]*sin(theta/2);
00284                 ket[b1][1] = e1[1]*cos(theta/2)-e0[0]*sin(theta/2);
00285             } else if (qc.data[g][0]==="h") {
00286                 for (int k=0; k<2; k++) {
00287                     ket[b0][k] = (e0[k] + e1[k])/sqrt(2);
00288                     ket[b1][k] = (e0[k] - e1[k])/sqrt(2);
00289                 }
00290             }
00291         }
00292     }
00293 }
00294
00295 } else if ( (qc.data[g][0]==="cx") or (qc.data[g][0]==="ch") or (qc.data[g][0]==="crx") ) {
00296     int s,t,l,h;
00297     s = stoi( qc.data[g][qc.data[g].size()-2] );
00298     t = stoi( qc.data[g][qc.data[g].size()-1] );
00299     if (s>t){
00300         h = s;
00301         l = t;
00302     } else {
00303         h = t;
00304         l = s;
00305     }
00306
00307     for (int i0=0; i0<pow(2,l); i0++){
00308         for (int il=0; il<pow(2,h-l-1); il++) {
00309             for (int i2=0; i2<pow(2,qc.nQubits-h-1); i2++) {
00310                 int b0,b1;
00311                 b0 = i0 + pow(2,l+1)*il + pow(2,h+1)*i2 + pow(2,s);
00312                 b1 = b0 + pow(2,t);
00313
00314                 vector<double> e0, e1;
00315                 e0 = ket[b0];
00316                 e1 = ket[b1];
00317
00318                 if (qc.data[g][0]==="cx") {
00319                     ket[b0] = e1;
00320                     ket[b1] = e0;
00321                 } else if (qc.data[g][0]==="ch") {
00322                     for (int k=0; k<2; k++) {
00323                         ket[b0][k] = (e0[k] + e1[k])/sqrt(2);
00324                         ket[b1][k] = (e0[k] - e1[k])/sqrt(2);
00325                     }
00326                 } else if (qc.data[g][0]==="crx") {
00327                     double theta = stof( qc.data[g][1] );
00328                     ket[b0][0] = e0[0]*cos(theta/2)+e1[1]*sin(theta/2);
00329                     ket[b0][1] = e0[1]*cos(theta/2)-e1[0]*sin(theta/2);
00330                     ket[b1][0] = e1[0]*cos(theta/2)+e0[1]*sin(theta/2);
00331                     ket[b1][1] = e1[1]*cos(theta/2)-e0[0]*sin(theta/2);
00332                 }
00333             }
00334         }
00335     }
00336 }
00337 }
00338 }
00339 }
00340
00341     return ket;
00342 }
00343
00344 vector<double> get_probs (QuantumCircuit qc) {
00345
00346     if(!qc.has_measurements()){
00347         ERROR("get_probs: The circuit should have a full set of measure gates");
00348     }
00349
00350     vector<vector<double> > ket;
00351     ket = simulate(qc);
00352
00353     vector<double> probs;
00354     for (int j=0; j<ket.size(); j++) {
00355
00356         probs.push_back( pow(ket[j][0],2) + pow(ket[j][1],2) );
00357
00358     }
00359
00360     return probs;
00361 }
00362

```

```

00363 public:
00364     QuantumCircuit qc;
00365     int shots;
00367
00368     Simulator (QuantumCircuit qc_in, int shots_in = 1024) {
00369         srand((unsigned)time(0)); //seed for rand() calculated from the epoch date
00370         qc = qc_in;
00371         shots = shots_in;
00372     }
00373
00374     vector<complex<double>> get_statevector () {
00375
00376         vector<vector<double>> ket;
00377         ket = simulate(qc);
00378         vector<complex<double>> complex_ket;
00379
00380         for (int j=0; j<ket.size(); j++) {
00381             complex<double> e (ket[j][0],ket[j][1]);
00382             complex_ket.push_back( e );
00383         }
00384
00385         return complex_ket;
00386     }
00387
00388     vector<string> get_memory () {
00389
00390         vector<double> probs;
00391         probs = get_probs(qc);
00392
00393         vector<string> memory;
00394
00395         for (int s=0; s<shots; s++) {
00396
00397             double cumu = 0;
00398             bool un = true;
00399             double r = double(rand()) / RAND_MAX;
00400             vector<char> bitstr (qc.nQubits,'0');
00401
00402             for (int j=0; j<probs.size(); j++) {
00403                 cumu += probs[j]; //this will add up to 1
00404                 if ((r <= cumu) && un) {
00405                     for (int w=0; w<bitstr.size(); w++) {
00406                         bool result = int(pow(2,w)) & j;
00407                         bitstr[qc.nQubits-1-w] = result?'1':'0';
00408                     }
00409                     string out(bitstr.begin(), bitstr.end());
00410                     memory.push_back( out );
00411                     un = false;
00412                 }
00413             }
00414         }
00415
00416         return memory; //e.g. <"10","10","10","10","10","10","10","10">
00417     }
00418
00419     map<string, int> get_counts () {
00420
00421         map<string, int> counts;
00422         vector<string> memory = get_memory();
00423         if(memory.size()>0){
00424             for (int s=0; s<shots; s++) {
00425                 counts[memory[s]] += 1; //aggregate by key/bitstr
00426             }
00427         }
00428
00429         return counts;
00430     }
00431
00432     string get_qiskit () {
00433         string qiskitPy;
00434
00435         if (qc.nBits==0){
00436             qiskitPy += "qc = QuantumCircuit (" + to_string(qc.nQubits) +")\n";
00437         } else {
00438             qiskitPy += "qc = QuantumCircuit (" + to_string(qc.nQubits) +", " + to_string(qc.nBits) +")\n";
00439         }
00440
00441         for (int g=0; g<qc.data.size(); g++) {
00442             if (qc.data[g][0]=="x"){
00443                 qiskitPy += "qc.x(" + qc.data[g][1] +")\n";
00444             } else if (qc.data[g][0]=="rx") {
00445                 qiskitPy += "qc.rx(" + qc.data[g][1] +", " + qc.data[g][2] +")\n";
00446             } else if (qc.data[g][0]=="h") {
00447                 qiskitPy += "qc.h(" + qc.data[g][1] +")\n";
00448             } else if (qc.data[g][0]=="cx") {
00449                 qiskitPy += "qc.cx(" + qc.data[g][1] +", " + qc.data[g][2] +")\n";
00450             }
00451         }
00452     }

```

```

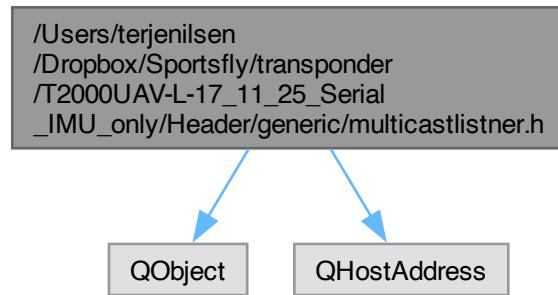
00450     } else if (qc.data[g][0]==="ch") {
00451         qiskitPy += "qc.ch(\""+qc.data[g][1]+"," +qc.data[g][2]+")\n";
00452     } else if (qc.data[g][0]==="crx") {
00453         qiskitPy += "qc.crx(\""+qc.data[g][1]+"," +qc.data[g][2]+"," +qc.data[g][3]+")\n";
00454     } else if (qc.data[g][0]==="m") {
00455         qiskitPy += "qc.measure(\""+qc.data[g][1]+"," +qc.data[g][2]+")\n";
00456     } else if (qc.data[g][0]==="init") {
00457         qiskitPy += "qc.initialize({\""+qc.data[g][2];
00458
00459         int initsize = stoi(qc.data[g][1]);
00460         for(int i=0; i<initsize-1; i++){
00461             qiskitPy += "," +qc.data[g][3+i];
00462         }
00463         qiskitPy += "})\n";
00464     }
00465 }
00466
00467     return qiskitPy;
00468 }
00469
00470     string get_qasm () {
00471         string qasm;
00472         // initial qasm header
00473         qasm += "OPENQASM 2.0;\ninclude \"qelib1.inc\";\n";
00474         // qreg
00475         qasm += "qreg q["+to_string(qc.nQubits)+"];\n";
00476         // creg
00477         if (qc.nBits!=0){ // maybe don't do this and always print it
00478             qasm += "creg c["+to_string(qc.nBits)+"];\n";
00479         }
00480         // gates
00481         for (int g=0; g<qc.data.size(); g++){
00482             if (qc.data[g][0]==="x"){
00483                 qasm += "x q["+qc.data[g][1]+"];\n";
00484             } else if (qc.data[g][0]==="rx") {
00485                 qasm += "rx(\""+qc.data[g][1]+") q["+qc.data[g][2]+"];\n";
00486             } else if (qc.data[g][0]==="h") {
00487                 qasm += "h q["+qc.data[g][1]+"];\n";
00488             } else if (qc.data[g][0]==="cx") {
00489                 qasm += "cx q["+qc.data[g][1]+"],q["+qc.data[g][2]+"];\n";
00490             } else if (qc.data[g][0]==="ch") {
00491                 qasm += "ch q["+qc.data[g][1]+"],q["+qc.data[g][2]+"];\n";
00492             } else if (qc.data[g][0]==="crx") {
00493                 qasm += "crx(\""+qc.data[g][1]+") q["+qc.data[g][2]+"],q["+qc.data[g][3]+"];\n";
00494             } else if (qc.data[g][0]==="m") {
00495                 qasm += "measure q["+qc.data[g][1]+"] -> c["+qc.data[g][2]+"];\n";
00496             }
00497         }
00498     }
00499     return qasm;
00500 }
00501
00502 };
00503 #endif

```

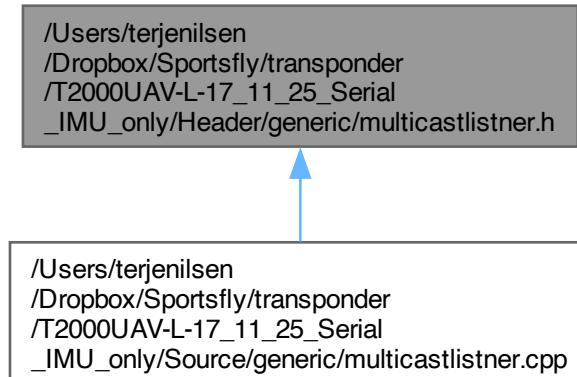
9.65 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_← 11_25_Serial_IMU_only/Header/generic/multicastlistner.h File Reference

```
#include <QObject>
#include <QHostAddress>
```

Include dependency graph for multicastlistner.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- class [MulticastListener](#)

9.66 multicastlistner.h

[Go to the documentation of this file.](#)

```
00001 #pragma once
00002
00003 #include <QObject>
00004 #include <QHostAddress>
00005
00006 class QSocketNotifier;
```

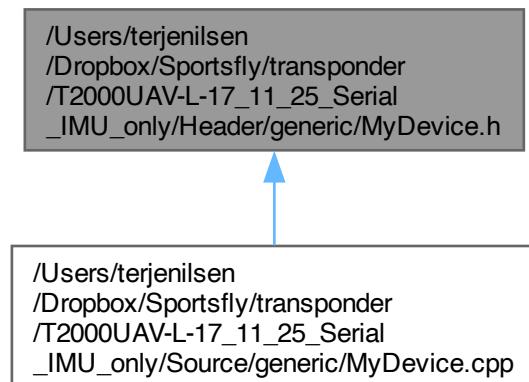
```

00007
00008 class MulticastListener : public QObject
00009 {
0010     Q_OBJECT
0011 public:
0012     explicit MulticastListener(QObject *parent = nullptr);
0013     ~MulticastListener() override;
0014
0015     bool start(const QString &groupAddr = QStringLiteral("239.255.0.1"),
0016                quint16 port = 4210);
0017     void stop();
0018
0019 signals:
0020     void messageReceived(const QString &message,
0021                          const QHostAddress &sender,
0022                          quint16 senderPort);
0023     void errorOccurred(const QString &error);
0024
0025 private slots:
0026     void onSocketActivated(int socket);
0027
0028 private:
0029     int m_fd = -1;
0030     QSockNotifier *m_notifier = nullptr;
0031     QString m_group;
0032     quint16 m_port = 0;
0033 };

```

9.67 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header/generic/MyDevice.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- class [MyDevice](#)

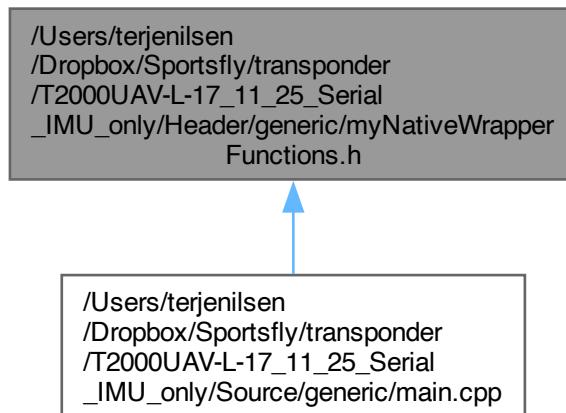
9.68 MyDevice.h

[Go to the documentation of this file.](#)

```
00001 #ifndef MYDEVICE_H
00002 #define MYDEVICE_H
00003
00004
00005 class MyDevice
00006 {
00007 public:
00008     MyDevice();
00009
00010
00011 public:
00012
00013 private:
00014
00015 };
00016
00017 #endif // MYDEVICE_H
```

9.69 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_← 11_25_Serial_IMU_only/Header/generic/myNativeWrapper← Functions.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- void [setIosParams \(\)](#)

9.69.1 Function Documentation

9.69.1.1 [setIosParams\(\)](#)

```
void setIosParams ()
```

9.70 myNativeWrapperFunctions.h

[Go to the documentation of this file.](#)

```
00001 /* */
00002 #ifndef SETIOSPARAMETERS_H
00003 #define SETIOSPARAMETERS_H
00004
00005 // Header to point toward Objective-C functions necessary for iOS deployment
00006 void setIosParams();
00007
00008 #endif // SETIOSPARAMETERS_H
```

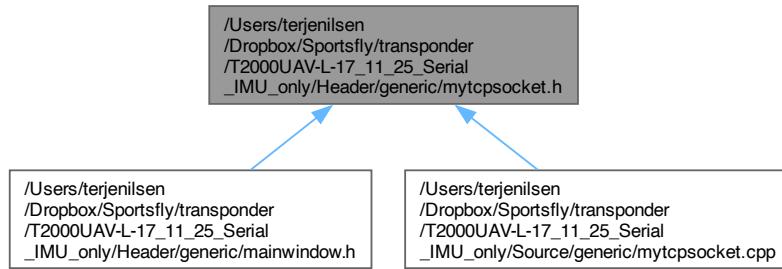
9.71 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_← 11_25_Serial_IMU_only/Header/generic/mytcpsocket.h File Reference

```
#include <QObject>
#include <QTcpSocket>
#include <QUdpSocket>
#include <QAbstractSocket>
#include <QDebug>
#include <QTimer>
#include <QPlainTextEdit>
#include <QDialog>
#include <QVBoxLayout>
#include <QLabel>
#include <QList>
#include <QHostAddress>
#include <QNetworkInterface>
#include "ekfNavINS.h"
#include "rotation_matrix.h"
#include "serialport.h"
#include "bleuart.h"
#include "mqttclient.h"
#include "tcpclient.h"
#include "ssdp.h"
#include <QSerialPort>
#include <QSerialPortInfo>
#include "ui_mainwindow_port_new.h"
```

Include dependency graph for mytcpsocket.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct [AltimeterData](#)
- class [NoButtonMessageBox](#)

Small frameless dialog for transient status messages.
- struct [PortEntry](#)

Convenience record for describing a serial port (used on macOS).
- class [MyTcpSocket](#)

Handles communication with:
- struct [MyTcpSocket::Callbacks](#)

Macros

- #define [USE_BT_IMU](#)
- #define [SCREEN](#) MainWindow_port_new
- #define [simGPS](#) false

Variables

- static constexpr char [STX](#) = 0x02
- static constexpr char [ETX](#) = 0x03

9.71.1 Macro Definition Documentation

9.71.1.1 SCREEN

```
#define SCREEN MainWindow_port_new
```

9.71.1.2 simGPS

```
#define simGPS false
```

9.71.1.3 USE_BT_IMU

```
#define USE_BT_IMU
```

9.71.2 Variable Documentation

9.71.2.1 ETX

```
char ETX = 0x03 [static], [constexpr]
```

9.71.2.2 STX

```
char STX = 0x02 [static], [constexpr]
```

9.72 mytcpsocket.h

[Go to the documentation of this file.](#)

```
00001 #ifndef MYTCP SOCKET_H
00002 #define MYTCP SOCKET_H
00003
00004 #include <QObject>
00005 #include <QTcpSocket>
00006 #include <QUdpSocket>
00007 #include <QAbstractSocket>
00008 #include <QDebug>
00009 #include <QTimer>
00010 #include <QPlainTextEdit>
00011 #include <QDialog>
00012 #include <QVBoxLayout>
00013 #include <QLabel>
00014 #include <QList>
00015 #include <QHostAddress>
00016 #include <QNetworkInterface>
00017
00018 #include "ekfNavINS.h"
00019 #include "rotation_matrix.h"
00020 #include "serialport.h"
00021 #include "bleuart.h"
00022 #include "mqttclient.h"
00023 #include "tcpclient.h"
00024 #include "ssdp.h"
00025
00026 // Look for an external IMU over Bluetooth
00027 #ifndef Q_OS_IOS
00028 #define USE_BT_IMU
00029 #else
00030 #undef USE_BT_IMU
00031 #endif
00032
00033 #ifdef Q_OS_IOS
00034 #undef Q_OS_MAC
00035 #endif
00036
00037 // Desktop platforms (non-Android, non-iOS) get Qt serial port support
00038 #if not defined(Q_OS_ANDROID) && not defined(Q_OS_IOS)
00039 #include <QSerialPort>
00040 #include <QSerialPortInfo>
00041 #endif
00042
00043 // -----
00044 // UI selection / simulation flags
00045 // -----
00046 #ifdef Q_OS_ANDROID
00047 // Android: main window layout
00048 #define SCREEN MainWindow_port_new
00049 #include "ui_mainwindow_port_new.h"
00050 #include "lockhelper.h"
```

```
00051 #define simGPS false
00052
00053 #elif defined(Q_OS_IOS)
00054 // If IOS for apple...
00055 #define SCREEN MainWindow_port_iPhone
00056 #include "ui_mainwindow_port_iPhone.h"
00057 #define simGPS false
00058
00059 #elif defined(Q_OS_MAC)
00060 // If MAC or PC screen...
00061 #include "ui_mainwindow_port_screen.h"
00062 #define SCREEN MainWindow_port_screen
00063 #define simGPS false
00064
00065 #else
00066 // Desktop: new layout, GPS simulation enabled
00067 #include "ui_mainwindow_port_new.h"
00068 #define SCREEN MainWindow_port_new
00069 #define simGPS false
00070 #endif
00071
00072 // STX/ETX used for framing protocols (if needed elsewhere)
00073 static constexpr char STX = 0x02;
00074 static constexpr char ETX = 0x03;
00075
00076
00077 #ifdef Q_OS_IOS
00078 #define ComBt void
00079 #define ComQt void
00080 #endif
00081
00082 // =====
00083 // NoButtonMessageBox
00084 // =====
00085 struct AltimeterData {
00086     float pressure;
00087     float temperature;
00088     float relative;
00089     float altitude;
00090 };
00091
00092 class NoButtonMessageBox : public QDialog {
00093     Q_OBJECT
00094 public:
00095     explicit NoButtonMessageBox(const QString &message, QWidget *parent = nullptr)
00096         : QDialog(parent)
00097     {
00098         setWindowFlags(Qt::FramelessWindowHint | Qt::Dialog);
00099         setAttribute(Qt::WA_TranslucentBackground);
00100         setModal(true);
00101
00102         auto *layout = new QVBoxLayout(this);
00103         m_label = new QLabel(message, this);
00104         m_label->setAlignment(Qt::AlignCenter);
00105         m_label->setStyleSheet(
00106             "QLabel { "
00107             "font-size: 18pt; "
00108             "color: white; "
00109             "background-color: #333; "
00110             "padding: 20px; "
00111             "border-radius: 12px; }"
00112         );
00113         layout->addWidget(m_label);
00114         setLayout(layout);
00115         resize(300, 100);
00116     }
00117
00118     void setText(const QString &text)
00119     {
00120         if (m_label)
00121             m_label->setText(text);
00122     }
00123
00124     private:
00125         QLabel *m_label = nullptr;
00126 };
00127
00128
00129 // =====
00130 // PortEntry
00131 // =====
00132
00133
00134
00135 struct PortEntry {
00136     QString serial;
00137     QString portName;
00138     QString systemLocation;
00139     QString description;
00140     QString manufacturer;
```

```

00152     quint16 vendorId = 0;
00153     quint16 productId = 0;
00154 };
00155
00156 // =====
00157 // MyTcpSocket
00158 // -----
00159
00160 class MyTcpSocket : public QObject
00161 {
00162     Q_OBJECT
00163
00164 public:
00165     explicit MyTcpSocket(QObject *parent = nullptr,
00166                           QPlainTextEdit *s = nullptr,
00167                           void (*retx)(void *, const char *, uint32_t) = nullptr,
00168                           void (*rety)(void *, bool use_imu) = nullptr);
00169     ~MyTcpSocket();
00170
00171     void ssdpConfig();
00172
00173     void readyWrite(char *data);
00174
00175     void doConnect();
00176
00177     void (*ret_imu)(void *, bool use_imu) = nullptr;
00178
00179     void (*ret_transponder)(void *, const char *data, uint32_t size) = nullptr;
00180
00181     void connected();
00182
00183     void connectedIMU();
00184
00185     void connectedRadar();
00186
00187     void connectedAltitude();
00188
00189     void setbacklit();
00190
00191     void setSerialPorts(QString imu, QString transponder, QString radar);
00192
00193     void handleUpdate(const std::string &ID, const std::string &value);
00194
00195     void parseAltimeterLine(const QString &line);
00196     AltimeterData Altimeter_data = {0,0,0,0};
00197
00198 #ifdef Q_OS_MAC
00199     static QVector<PortEntry> listSerialPortsDetailed();
00200
00201     static QMap<QString, QString> serialToPortMap(bool useSystemLocation = true);
00202
00203     QString findPort(QString targetSerial);
00204
00205     QMap<QString, QString> map;
00206 #endif
00207
00208     QString m_address = "239.255.0.1";
00209     quint16 MCAST_PORT = 4210;
00210
00211 #if not defined(Q_OS_ANDROID) && not defined(Q_OS_IOS)
00212     int com_setup(QSerialPort *com_port, QString sport);
00213
00214     QSerialPort *port = nullptr;
00215     QSerialPort *lidar = nullptr;
00216
00217     QList<QSerialPortInfo> serialport;
00218 #endif
00219     SsdpDiscoverer *disc = nullptr;
00220
00221     QString sport;
00222
00223     QPlainTextEdit *text = nullptr;
00224
00225     QString _transponder_id;
00226     QString _radar_id;
00227     QString _IMU_id;
00228
00229 // -----
00230 // MQTT state
00231 // -----
00232     std::string SERVER_ADDRESS;
00233     std::string CLIENT_ID;
00234     MqttClient *mqtt = nullptr;
00235
00236     bool m_has_MQTT      = false;
00237     bool m_has_MQTT_gyro = false;

```

```
00337     bool m_has_MQTT_accel      = false;
00338     bool m_has_MQTT_vsi       = false;
00339     bool m_has_MQTT_heading    = false;
00340     bool m_has_MQTT_airspeed   = false;
00341     bool m_has_MQTT_presssure = false;
00342
00343 #ifdef Q_OS_ANDROID
00344     QJniObject *someJavaObject = nullptr;
00345     QJniObject *imuJavaObject = nullptr;
00346 #else
00347     // -----
00348     // iOS / desktop: generic callback bridge (e.g. for Swift interop).
00349     // -----
00350
00351     typedef struct Callbacks
00352     {
00353         void *classPtr;
00354         void (*callback)(void *);
00355     } Callbacks;
00356
00357     Callbacks *callbacks = nullptr;
00358
00359     void CallSwiftMemberFromC(void *classPtr, void (*callback)(void *)) {
00360         callbacks->classPtr = classPtr;
00361         callbacks->callback = callback;
00362
00363         std::function<void()> actualCallback = [&]() {
00364             callbacks->callback(callbacks->classPtr);
00365         };
00366         actualCallback();
00367     }
00368 #endif
00369
00370     // -----
00371     // IMU / sensor data (decoded values)
00372     // -----
00373     QString imuData;
00374 #ifdef Q_OS_MAC
00375     bool Transponderstat = true;
00376 #else
00377     bool Transponderstat = false;
00378 #endif
00379     bool Altitudestat = false;
00380
00381     bool Radarstat = false;
00382     float rPos     = 0.0f;
00383     float rSpeed   = 0.0f;
00384     float rDist    = 0.0f;
00385
00386     bool IMUconnected = false;
00387     bool m_external   = true;
00388     bool m_imu_setup_done = false;
00389
00390     double m_pmeasure_QNH = -10000;
00391     bool TransponderstatWithBarometer = false;
00392
00393     // IMU output variables (SI units where possible)
00394     QString FromID;
00395     double AccX = 0.0;
00396     double AccY = 0.0;
00397     double AccZ = 0.0;
00398     double G    = Gfix;
00399
00400     double AsX = 0.0;
00401     double AsY = 0.0;
00402     double AsZ = 0.0;
00403
00404     double AngleX = 0.0;
00405     double AngleY = 0.0;
00406     double AngleZ = 0.001;
00407
00408     double HX = 0.0;
00409     double HY = 0.0;
00410     double HZ = 0.0;
00411
00412     quint16 VER = 0.0;
00413     double Temp = -100.0;
00414
00415     // GPS / altitude
00416     double m_altitude      = -100.0;
00417     double m_latitude       = 0.0;
00418     double m_longitude      = 0.0;
00419
00420     // GPS velocity vector
00421     double m_gpsspeed     = 0.0;
00422     double m_gpsbearing   = 0.0;
00423     double m_vel_N        = 0.0;
```

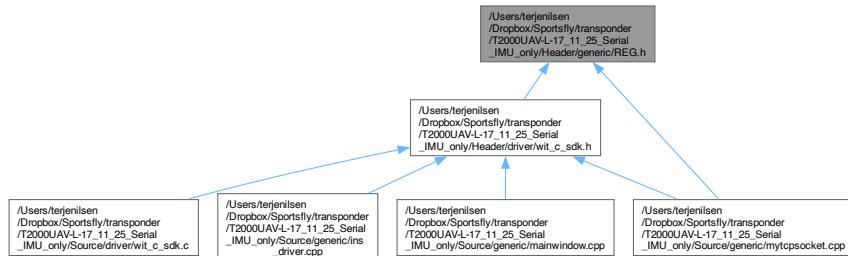
```

00433     double m_vel_E      = 0.0;
00434     double m_vel_D      = 0.0;
00435     bool   m_vel_active = false;
00436
00438     double m_pmeasure_alt  = 0.0;
00440     double m_pmeasure      = 0.0;
00442     double m_pressure_raw  = 0.0;
00443
00445     double m_speed       = 0.0;
00446     double Downn_Speed   = 0.0;
00447
00448     int Orient        = 0;
00449
00450
00451     bool use_ins_only    = false;
00452
00453
00454
00455 private:
00456     QUdpSocket *m_multicastSender = nullptr;
00457
00458 // -----
00459 // Hardware communication backends
00460 // -----
00461 #ifndef Q_OS_IOS
00462     ComQt *TransponderSerPort = nullptr;
00463     ComQt *RadarSerPort      = nullptr;
00464     ComQt *INSSerPort        = nullptr;
00465     ComBt *bluetoothPort    = nullptr;
00466 #endif
00467     QString m_imu_address = "";
00468     QString m_radar_address = "";
00469     QString m_transponder_address = "";
00470     QString m_altimeter_address = "";
00471
00472     QTcpSocket *m_imuClient = nullptr;
00473     QTcpSocket *m_radarClient = nullptr;
00474     QTcpSocket *m_transponderClient = nullptr;
00475     QTcpSocket *m_altimeterClient = nullptr;
00476
00477 signals:
00478     void sendMessage(const QString &message);
00479
00480 public slots:
00481     void doTransponder();
00482
00483     void doStart();
00484
00485     //static void doIMU(void *parent, const char *data, uint32_t length);
00486     static void parseIMU(void *parent, uint32_t uiReg, uint16_t sReg[]);
00487
00488     static void doRadar(void *parent, const char *data, uint32_t length);
00489
00490     // void SensorUartSend(uint8_t *p_data, uint32_t uiSize); // legacy, unused
00491
00492 private:
00493 // -----
00494 // Internal timers
00495 // -----
00496     QTimer *timer        = nullptr;
00497     QTimer *timerTRANS  = nullptr;
00498     QTimer *timerIMU    = nullptr;
00499     QTimer *java         = nullptr;
00500     QTimer *timerStart  = nullptr;
00501
00502     QObject *parent      = nullptr;
00503
00504     int adapterFromUserSelection() const;
00505     int currentAdapterIndex = 0;
00506
00507     void reactOnSocketError(const QString &error);
00508
00509     QString localName;
00510
00511 };
00512
00513 #endif // MYTCP SOCKET_H

```

9.73 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header/generic/REG.h File Reference

This graph shows which files directly or indirectly include this file:



Macros

- #define REGSIZE 0x90
- #define SAVE 0x00
- #define CALSW 0x01
- #define RSW 0x02
- #define RRATE 0x03
- #define BAUD 0x04
- #define AXOFFSET 0x05
- #define AYOFFSET 0x06
- #define AZOFFSET 0x07
- #define GXOFFSET 0x08
- #define GYOFFSET 0x09
- #define GZOFFSET 0x0a
- #define HXOFFSET 0x0b
- #define HYOFFSET 0x0c
- #define HZOFFSET 0x0d
- #define D0MODE 0x0e
- #define D1MODE 0x0f
- #define D2MODE 0x10
- #define D3MODE 0x11
- #define D0PWMH 0x12
- #define D1PWMH 0x13
- #define D2PWMH 0x14
- #define D3PWMH 0x15
- #define D0PWMT 0x16
- #define D1PWMT 0x17
- #define D2PWMT 0x18
- #define D3PWMT 0x19
- #define IICADDR 0x1a
- #define LEDOFF 0x1b
- #define MAGRANGX 0x1c
- #define MAGRANGY 0x1d
- #define MAGRANGZ 0x1e
- #define BANDWIDTH 0x1f
- #define GYRORANGE 0x20

- #define ACCRANGE 0x21
- #define SLEEP 0x22
- #define ORIENT 0x23
- #define AXIS6 0x24
- #define FILTK 0x25
- #define GPSBAUD 0x26
- #define READADDR 0x27
- #define BWSCALE 0x28
- #define MOVETHR 0x28
- #define MOVESTA 0x29
- #define ACCFILT 0x2A
- #define GYROFILT 0x2b
- #define MAGFILT 0x2c
- #define POWONSEND 0x2d
- #define VERSION 0x2e
- #define CCBW 0x2f
- #define YYMM 0x30
- #define DDHH 0x31
- #define MMSS 0x32
- #define MS 0x33
- #define AX 0x34
- #define AY 0x35
- #define AZ 0x36
- #define GX 0x37
- #define GY 0x38
- #define GZ 0x39
- #define HXi 0x3a
- #define HYi 0x3b
- #define HZi 0x3c
- #define Roll 0x3d
- #define Pitch 0x3e
- #define Yaw 0x3f
- #define TEMP 0x40
- #define LRoll 0x3d
- #define HRoll 0x3e
- #define LPitch 0x3f
- #define HPitch 0x40
- #define LYaw 0x41
- #define HYaw 0x42
- #define TEMP905x 0x43
- #define D0Status 0x41
- #define D1Status 0x42
- #define D2Status 0x43
- #define D3Status 0x44
- #define PressureL 0x45
- #define PressureH 0x46
- #define HeightL 0x47
- #define HeightH 0x48
- #define LonL 0x49
- #define LonH 0x4a
- #define LatL 0x4b
- #define LatH 0x4c
- #define GPSHeight 0x4d
- #define GPSVX_L 0x4E
- #define GPSVX_H 0x4F

- #define GPSVY_L 0x50
- #define GPSVY_H 0x51
- #define GPSVZ_L 0x52
- #define GPSVZ_H 0x53
- #define GPSYAW 0x4e
- #define GPSVL 0x4f
- #define GPSVH 0x50
- #define q0 0x51
- #define q1 0x52
- #define q2 0x53
- #define q3 0x54
- #define SVNUM 0x55
- #define PDOP 0x56
- #define HDOP 0x57
- #define VDOP 0x58
- #define DELAYT 0x59
- #define XMIN 0x5a
- #define XMAX 0x5b
- #define BATVAL 0x5c
- #define ALARMPIN 0x5d
- #define YMIN 0x5e
- #define YMAX 0x5f
- #define GYROZSCALE 0x60
- #define GYROCALITHR 0x61
- #define ALARMLEVEL 0x62
- #define GYROCALTIME 0x63
- #define REFROLL 0x64
- #define REFPITCH 0x65
- #define REFYAW 0x66
- #define GPSTYPE 0x67
- #define TRIGTIME 0x68
- #define KEY 0x69
- #define WERROR 0x6a
- #define TIMEZONE 0x6b
- #define CALICNT 0x6c
- #define WZCNT 0x6d
- #define WZTIME 0x6e
- #define WZSTATIC 0x6f
- #define ACCSENSOR 0x70
- #define GYROSENSOR 0x71
- #define MAGSENSOR 0x72
- #define PRESSENSOR 0x73
- #define MODDELAY 0x74
- #define ANGLEAXIS 0x75
- #define XRSCALE 0x76
- #define YRSCALE 0x77
- #define ZRSCALE 0x78
- #define XREFROLL 0x79
- #define YREFPITCH 0x7a
- #define ZREFYAW 0x7b
- #define ANGXOFFSET 0x7c
- #define ANGYOFFSET 0x7d
- #define ANGZOFFSET 0x7e
- #define NUMBERID1 0x7f
- #define NUMBERID2 0x80

- #define NUMBERID3 0x81
- #define NUMBERID4 0x82
- #define NUMBERID5 0x83
- #define NUMBERID6 0x84
- #define XA85PSCALE 0x85
- #define XA85NSCALE 0x86
- #define YA85PSCALE 0x87
- #define YA85NSCALE 0x88
- #define XA30PSCALE 0x89
- #define XA30NSCALE 0x8a
- #define YA30PSCALE 0x8b
- #define YA30NSCALE 0x8c
- #define CHIPIDL 0x8D
- #define CHIPIDH 0x8E
- #define REGINITFLAG REGSIZE-1
- #define ALGRITHM9 0
- #define ALGRITHM6 1
- #define NORMAL 0x00
- #define CALGYROACC 0x01
- #define CALMAG 0x02
- #define CALALTITUDE 0x03
- #define CALANGLEZ 0x04
- #define CALACCL 0x05
- #define CALACCR 0x06
- #define CALMAGMM 0x07
- #define CALREFANGLE 0x08
- #define CALMAG2STEP 0x09
- #define CALHEXAHEDRON 0x12
- #define WIT_TIME 0x50
- #define WIT_ACC 0x51
- #define WIT_GYRO 0x52
- #define WIT_ANGLE 0x53
- #define WIT_MAGNETIC 0x54
- #define WIT_DPORT 0x55
- #define WIT_PRESS 0x56
- #define WIT_GPS 0x57
- #define WIT_VELOCITY 0x58
- #define WIT_QUATER 0x59
- #define WIT_GSA 0x5A
- #define WIT_REGVALUE 0x5F
- #define RSW_TIME 0x01
- #define RSW_ACC 0x02
- #define RSW_GYRO 0x04
- #define RSW_ANGLE 0x08
- #define RSW_MAG 0x10
- #define RSW_PORT 0x20
- #define RSW_PRESS 0x40
- #define RSW_GPS 0x80
- #define RSW_V 0x100
- #define RSW_Q 0x200
- #define RSW_GSA 0x400
- #define RSW_MASK 0xffff
- #define RRATE_NONE 0x0d
- #define RRATE_02HZ 0x01
- #define RRATE_05HZ 0x02

- #define RRATE_1HZ 0x03
- #define RRATE_2HZ 0x04
- #define RRATE_5HZ 0x05
- #define RRATE_10HZ 0x06
- #define RRATE_20HZ 0x07
- #define RRATE_50HZ 0x08
- #define RRATE_100HZ 0x09
- #define RRATE_125HZ 0x0a
- #define RRATE_200HZ 0x0b
- #define RRATE_ONCE 0x0c
- #define WIT_BAUD_4800 1
- #define WIT_BAUD_9600 2
- #define WIT_BAUD_19200 3
- #define WIT_BAUD_38400 4
- #define WIT_BAUD_57600 5
- #define WIT_BAUD_115200 6
- #define WIT_BAUD_230400 7
- #define WIT_BAUD_460800 8
- #define WIT_BAUD_921600 9
- #define CAN_BAUD_1000000 0
- #define CAN_BAUD_800000 1
- #define CAN_BAUD_500000 2
- #define CAN_BAUD_400000 3
- #define CAN_BAUD_250000 4
- #define CAN_BAUD_200000 5
- #define CAN_BAUD_125000 6
- #define CAN_BAUD_100000 7
- #define CAN_BAUD_80000 8
- #define CAN_BAUD_50000 9
- #define CAN_BAUD_40000 10
- #define CAN_BAUD_20000 11
- #define CAN_BAUD_10000 12
- #define CAN_BAUD_5000 13
- #define CAN_BAUD_3000 14
- #define KEY_UNLOCK 0xB588
- #define SAVE_PARAM 0x00
- #define SAVE_SWRST 0xFF
- #define ORIENT_HERIZONE 0
- #define ORIENT_VERTICLE 1
- #define BANDWIDTH_256HZ 0
- #define BANDWIDTH_184HZ 1
- #define BANDWIDTH_94HZ 2
- #define BANDWIDTH_44HZ 3
- #define BANDWIDTH_21HZ 4
- #define BANDWIDTH_10HZ 5
- #define BANDWIDTH_5HZ 6

9.73.1 Macro Definition Documentation

9.73.1.1 ACCFILT

```
#define ACCFILT 0x2A
```

9.73.1.2 ACCRANGE

```
#define ACCRANGE 0x21
```

9.73.1.3 ACCSENSOR

```
#define ACCSENSOR 0x70
```

9.73.1.4 ALARMLEVEL

```
#define ALARMLEVEL 0x62
```

9.73.1.5 ALARMPIN

```
#define ALARMPIN 0x5d
```

9.73.1.6 ALGRITHM6

```
#define ALGRITHM6 1
```

9.73.1.7 ALGRITHM9

```
#define ALGRITHM9 0
```

9.73.1.8 ANGLEAXIS

```
#define ANGLEAXIS 0x75
```

9.73.1.9 ANGXOFFSET

```
#define ANGXOFFSET 0x7c
```

9.73.1.10 ANGYOFFSET

```
#define ANGYOFFSET 0x7d
```

9.73.1.11 ANGZOFFSET

```
#define ANGZOFFSET 0x7e
```

9.73.1.12 AX

```
#define AX 0x34
```

9.73.1.13 AXIS6

```
#define AXIS6 0x24
```

9.73.1.14 AXOFFSET

```
#define AXOFFSET 0x05
```

9.73.1.15 AY

```
#define AY 0x35
```

9.73.1.16 AYOFFSET

```
#define AYOFFSET 0x06
```

9.73.1.17 AZ

```
#define AZ 0x36
```

9.73.1.18 AZOFFSET

```
#define AZOFFSET 0x07
```

9.73.1.19 BANDWIDTH

```
#define BANDWIDTH 0x1f
```

9.73.1.20 BANDWIDTH_10HZ

```
#define BANDWIDTH_10HZ 5
```

9.73.1.21 BANDWIDTH_184HZ

```
#define BANDWIDTH_184HZ 1
```

9.73.1.22 BANDWIDTH_21HZ

```
#define BANDWIDTH_21HZ 4
```

9.73.1.23 BANDWIDTH_256HZ

```
#define BANDWIDTH_256HZ 0
```

9.73.1.24 BANDWIDTH_44HZ

```
#define BANDWIDTH_44HZ 3
```

9.73.1.25 BANDWIDTH_5HZ

```
#define BANDWIDTH_5HZ 6
```

9.73.1.26 BANDWIDTH_94HZ

```
#define BANDWIDTH_94HZ 2
```

9.73.1.27 BATVAL

```
#define BATVAL 0x5c
```

9.73.1.28 BAUD

```
#define BAUD 0x04
```

9.73.1.29 BWSCALE

```
#define BWSCALE 0x28
```

9.73.1.30 CALACCL

```
#define CALACCL 0x05
```

9.73.1.31 CALACCR

```
#define CALACCR 0x06
```

9.73.1.32 CALALTITUDE

```
#define CALALTITUDE 0x03
```

9.73.1.33 CALANGLEZ

```
#define CALANGLEZ 0x04
```

9.73.1.34 CALGYROACC

```
#define CALGYROACC 0x01
```

9.73.1.35 CALHEXAHEDRON

```
#define CALHEXAHEDRON 0x12
```

9.73.1.36 CALICNT

```
#define CALICNT 0x6c
```

9.73.1.37 CALMAG

```
#define CALMAG 0x02
```

9.73.1.38 CALMAG2STEP

```
#define CALMAG2STEP 0x09
```

9.73.1.39 CALMAGMM

```
#define CALMAGMM 0x07
```

9.73.1.40 CALREFANGLE

```
#define CALREFANGLE 0x08
```

9.73.1.41 CALSW

```
#define CALSW 0x01
```

9.73.1.42 CAN_BAUD_10000

```
#define CAN_BAUD_10000 12
```

9.73.1.43 CAN_BAUD_100000

```
#define CAN_BAUD_100000 7
```

9.73.1.44 CAN_BAUD_1000000

```
#define CAN_BAUD_1000000 0
```

9.73.1.45 CAN_BAUD_125000

```
#define CAN_BAUD_125000 6
```

9.73.1.46 CAN_BAUD_20000

```
#define CAN_BAUD_20000 11
```

9.73.1.47 CAN_BAUD_200000

```
#define CAN_BAUD_200000 5
```

9.73.1.48 CAN_BAUD_250000

```
#define CAN_BAUD_250000 4
```

9.73.1.49 CAN_BAUD_3000

```
#define CAN_BAUD_3000 14
```

9.73.1.50 CAN_BAUD_40000

```
#define CAN_BAUD_40000 10
```

9.73.1.51 CAN_BAUD_400000

```
#define CAN_BAUD_400000 3
```

9.73.1.52 CAN_BAUD_5000

```
#define CAN_BAUD_5000 13
```

9.73.1.53 CAN_BAUD_50000

```
#define CAN_BAUD_50000 9
```

9.73.1.54 CAN_BAUD_500000

```
#define CAN_BAUD_500000 2
```

9.73.1.55 CAN_BAUD_80000

```
#define CAN_BAUD_80000 8
```

9.73.1.56 CAN_BAUD_800000

```
#define CAN_BAUD_800000 1
```

9.73.1.57 CCBW

```
#define CCBW 0x2f
```

9.73.1.58 CHIPIDH

```
#define CHIPIDH 0x8E
```

9.73.1.59 CHIPIDL

```
#define CHIPIDL 0x8D
```

9.73.1.60 D0MODE

```
#define D0MODE 0x0e
```

9.73.1.61 D0PWMH

```
#define D0PWMH 0x12
```

9.73.1.62 D0PWMT

```
#define D0PWMT 0x16
```

9.73.1.63 D0Status

```
#define D0Status 0x41
```

9.73.1.64 D1MODE

```
#define D1MODE 0x0f
```

9.73.1.65 D1PWMH

```
#define D1PWMH 0x13
```

9.73.1.66 D1PWMT

```
#define D1PWMT 0x17
```

9.73.1.67 D1Status

```
#define D1Status 0x42
```

9.73.1.68 D2MODE

```
#define D2MODE 0x10
```

9.73.1.69 D2PWMH

```
#define D2PWMH 0x14
```

9.73.1.70 D2PWMT

```
#define D2PWMT 0x18
```

9.73.1.71 D2Status

```
#define D2Status 0x43
```

9.73.1.72 D3MODE

```
#define D3MODE 0x11
```

9.73.1.73 D3PWMH

```
#define D3PWMH 0x15
```

9.73.1.74 D3PWMT

```
#define D3PWMT 0x19
```

9.73.1.75 D3Status

```
#define D3Status 0x44
```

9.73.1.76 DDHH

```
#define DDHH 0x31
```

9.73.1.77 DELAYT

```
#define DELAYT 0x59
```

9.73.1.78 FILTK

```
#define FILTK 0x25
```

9.73.1.79 GPSBAUD

```
#define GPSBAUD 0x26
```

9.73.1.80 GPSHeight

```
#define GPSHeight 0x4d
```

9.73.1.81 GPSTYPE

```
#define GPSTYPE 0x67
```

9.73.1.82 GPSVH

```
#define GPSVH 0x50
```

9.73.1.83 GPSVL

```
#define GPSVL 0x4f
```

9.73.1.84 GPSVX_H

```
#define GPSVX_H 0x4F
```

9.73.1.85 GPSVX_L

```
#define GPSVX_L 0x4E
```

9.73.1.86 GPSVY_H

```
#define GPSVY_H 0x51
```

9.73.1.87 GPSVY_L

```
#define GPSVY_L 0x50
```

9.73.1.88 GPSVZ_H

```
#define GPSVZ_H 0x53
```

9.73.1.89 GPSVZ_L

```
#define GPSVZ_L 0x52
```

9.73.1.90 GPSYAW

```
#define GPSYAW 0x4e
```

9.73.1.91 GX

```
#define GX 0x37
```

9.73.1.92 GXOFFSET

```
#define GXOFFSET 0x08
```

9.73.1.93 GY

```
#define GY 0x38
```

9.73.1.94 GYOFFSET

```
#define GYOFFSET 0x09
```

9.73.1.95 GYROCALTHR

```
#define GYROCALTHR 0x61
```

9.73.1.96 GYROCALTIME

```
#define GYROCALTIME 0x63
```

9.73.1.97 GYROFILT

```
#define GYROFILT 0x2b
```

9.73.1.98 GYRORANGE

```
#define GYRORANGE 0x20
```

9.73.1.99 GYROSENSOR

```
#define GYROSENSOR 0x71
```

9.73.1.100 GYROZSCALE

```
#define GYROZSCALE 0x60
```

9.73.1.101 GZ

```
#define GZ 0x39
```

9.73.1.102 GZOFFSET

```
#define GZOFFSET 0x0a
```

9.73.1.103 HDOP

```
#define HDOP 0x57
```

9.73.1.104 HeightH

```
#define HeightH 0x48
```

9.73.1.105 HeightL

```
#define HeightL 0x47
```

9.73.1.106 HPitch

```
#define HPitch 0x40
```

9.73.1.107 HRoll

```
#define HRoll 0x3e
```

9.73.1.108 HXi

```
#define HXi 0x3a
```

9.73.1.109 HXOFFSET

```
#define HXOFFSET 0x0b
```

9.73.1.110 HYaw

```
#define HYaw 0x42
```

9.73.1.111 HYi

```
#define HYi 0x3b
```

9.73.1.112 HYOFFSET

```
#define HYOFFSET 0x0c
```

9.73.1.113 HZi

```
#define HZi 0x3c
```

9.73.1.114 HZOFFSET

```
#define HZOFFSET 0x0d
```

9.73.1.115 IICADDR

```
#define IICADDR 0x1a
```

9.73.1.116 KEY

```
#define KEY 0x69
```

9.73.1.117 KEY_UNLOCK

```
#define KEY_UNLOCK 0xB588
```

9.73.1.118 LatH

```
#define LatH 0x4c
```

9.73.1.119 LatL

```
#define LatL 0x4b
```

9.73.1.120 LEDOFF

```
#define LEDOFF 0x1b
```

9.73.1.121 LonH

```
#define LonH 0x4a
```

9.73.1.122 LonL

```
#define LonL 0x49
```

9.73.1.123 LPitch

```
#define LPitch 0x3f
```

9.73.1.124 LRoll

```
#define LRoll 0x3d
```

9.73.1.125 LYaw

```
#define LYaw 0x41
```

9.73.1.126 MAGFILT

```
#define MAGFILT 0x2c
```

9.73.1.127 MAGRANGX

```
#define MAGRANGX 0x1c
```

9.73.1.128 MAGRANGY

```
#define MAGRANGY 0x1d
```

9.73.1.129 MAGRANGZ

```
#define MAGRANGZ 0x1e
```

9.73.1.130 MAGSENSOR

```
#define MAGSENSOR 0x72
```

9.73.1.131 MMSS

```
#define MMSS 0x32
```

9.73.1.132 MODDELAY

```
#define MODDELAY 0x74
```

9.73.1.133 MOVESTA

```
#define MOVESTA 0x29
```

9.73.1.134 MOVETHR

```
#define MOVETHR 0x28
```

9.73.1.135 MS

```
#define MS 0x33
```

9.73.1.136 NORMAL

```
#define NORMAL 0x00
```

9.73.1.137 NUMBERID1

```
#define NUMBERID1 0x7f
```

9.73.1.138 NUMBERID2

```
#define NUMBERID2 0x80
```

9.73.1.139 NUMBERID3

```
#define NUMBERID3 0x81
```

9.73.1.140 NUMBERID4

```
#define NUMBERID4 0x82
```

9.73.1.141 NUMBERID5

```
#define NUMBERID5 0x83
```

9.73.1.142 NUMBERID6

```
#define NUMBERID6 0x84
```

9.73.1.143 ORIENT

```
#define ORIENT 0x23
```

9.73.1.144 ORIENT_HERIZONE

```
#define ORIENT_HERIZONE 0
```

9.73.1.145 ORIENT_VERTICLE

```
#define ORIENT_VERTICLE 1
```

9.73.1.146 PDOP

```
#define PDOP 0x56
```

9.73.1.147 Pitch

```
#define Pitch 0x3e
```

9.73.1.148 POWONSEND

```
#define POWONSEND 0x2d
```

9.73.1.149 PRESSENSOR

```
#define PRESSENSOR 0x73
```

9.73.1.150 PressureH

```
#define PressureH 0x46
```

9.73.1.151 PressureL

```
#define PressureL 0x45
```

9.73.1.152 q0

```
#define q0 0x51
```

9.73.1.153 q1

```
#define q1 0x52
```

9.73.1.154 q2

```
#define q2 0x53
```

9.73.1.155 q3

```
#define q3 0x54
```

9.73.1.156 READADDR

```
#define READADDR 0x27
```

9.73.1.157 REFPITCH

```
#define REFPITCH 0x65
```

9.73.1.158 REFROLL

```
#define REFROLL 0x64
```

9.73.1.159 REFYAW

```
#define REFYAW 0x66
```

9.73.1.160 REGINITFLAG

```
#define REGINITFLAG REGSIZE-1
```

9.73.1.161 REGSIZE

```
#define REGSIZE 0x90
```

9.73.1.162 Roll

```
#define Roll 0x3d
```

9.73.1.163 RRATE

```
#define RRATE 0x03
```

9.73.1.164 RRATE_02HZ

```
#define RRATE_02HZ 0x01
```

9.73.1.165 RRATE_05HZ

```
#define RRATE_05HZ 0x02
```

9.73.1.166 RRATE_100HZ

```
#define RRATE_100HZ 0x09
```

9.73.1.167 RRATE_10HZ

```
#define RRATE_10HZ 0x06
```

9.73.1.168 RRATE_125HZ

```
#define RRATE_125HZ 0x0a
```

9.73.1.169 RRATE_1HZ

```
#define RRATE_1HZ 0x03
```

9.73.1.170 RRATE_200HZ

```
#define RRATE_200HZ 0x0b
```

9.73.1.171 RRATE_20HZ

```
#define RRATE_20HZ 0x07
```

9.73.1.172 RRATE_2HZ

```
#define RRATE_2HZ 0x04
```

9.73.1.173 RRATE_50HZ

```
#define RRATE_50HZ 0x08
```

9.73.1.174 RRATE_5HZ

```
#define RRATE_5HZ 0x05
```

9.73.1.175 RRATE_NONE

```
#define RRATE_NONE 0x0d
```

RRATE

9.73.1.176 RRATE_ONCE

```
#define RRATE_ONCE 0x0c
```

9.73.1.177 RSW

```
#define RSW 0x02
```

9.73.1.178 RSW_ACC

```
#define RSW_ACC 0x02
```

9.73.1.179 RSW_ANGLE

```
#define RSW_ANGLE 0x08
```

9.73.1.180 RSW_GPS

```
#define RSW_GPS 0x80
```

9.73.1.181 RSW_GSA

```
#define RSW_GSA 0x400
```

9.73.1.182 RSW_GYRO

```
#define RSW_GYRO 0x04
```

9.73.1.183 RSW_MAG

```
#define RSW_MAG 0x10
```

9.73.1.184 RSW_MASK

```
#define RSW_MASK 0xffff
```

9.73.1.185 RSW_PORT

```
#define RSW_PORT 0x20
```

9.73.1.186 RSW_PRESS

```
#define RSW_PRESS 0x40
```

9.73.1.187 RSW_Q

```
#define RSW_Q 0x200
```

9.73.1.188 RSW_TIME

```
#define RSW_TIME 0x01
```

9.73.1.189 RSW_V

```
#define RSW_V 0x100
```

9.73.1.190 SAVE

```
#define SAVE 0x00
```

9.73.1.191 SAVE_PARAM

```
#define SAVE_PARAM 0x00
```

9.73.1.192 SAVE_SWRST

```
#define SAVE_SWRST 0xFF
```

9.73.1.193 SLEEP

```
#define SLEEP 0x22
```

9.73.1.194 SVNUM

```
#define SVNUM 0x55
```

9.73.1.195 TEMP

```
#define TEMP 0x40
```

9.73.1.196 TEMP905x

```
#define TEMP905x 0x43
```

9.73.1.197 TIMEZONE

```
#define TIMEZONE 0x6b
```

9.73.1.198 TRIGTIME

```
#define TRIGTIME 0x68
```

9.73.1.199 VDOP

```
#define VDOP 0x58
```

9.73.1.200 VERSION

```
#define VERSION 0x2e
```

9.73.1.201 WERROR

```
#define WERROR 0x6a
```

9.73.1.202 WIT_ACC

```
#define WIT_ACC 0x51
```

9.73.1.203 WIT_ANGLE

```
#define WIT_ANGLE 0x53
```

9.73.1.204 WIT_BAUD_115200

```
#define WIT_BAUD_115200 6
```

9.73.1.205 WIT_BAUD_19200

```
#define WIT_BAUD_19200 3
```

9.73.1.206 WIT_BAUD_230400

```
#define WIT_BAUD_230400 7
```

9.73.1.207 WIT_BAUD_38400

```
#define WIT_BAUD_38400 4
```

9.73.1.208 WIT_BAUD_460800

```
#define WIT_BAUD_460800 8
```

9.73.1.209 WIT_BAUD_4800

```
#define WIT_BAUD_4800 1
```

9.73.1.210 WIT_BAUD_57600

```
#define WIT_BAUD_57600 5
```

9.73.1.211 WIT_BAUD_921600

```
#define WIT_BAUD_921600 9
```

9.73.1.212 WIT_BAUD_9600

```
#define WIT_BAUD_9600 2
```

9.73.1.213 WIT_DPORT

```
#define WIT_DPORT 0x55
```

9.73.1.214 WIT_GPS

```
#define WIT_GPS 0x57
```

9.73.1.215 WIT_GSA

```
#define WIT_GSA 0x5A
```

9.73.1.216 WIT_GYRO

```
#define WIT_GYRO 0x52
```

9.73.1.217 WIT_MAGNETIC

```
#define WIT_MAGNETIC 0x54
```

9.73.1.218 WIT_PRESS

```
#define WIT_PRESS 0x56
```

9.73.1.219 WIT_QUATER

```
#define WIT_QUATER 0x59
```

9.73.1.220 WIT_REGVALUE

```
#define WIT_REGVALUE 0x5F
```

9.73.1.221 WIT_TIME

```
#define WIT_TIME 0x50
```

9.73.1.222 WIT_VELOCITY

```
#define WIT_VELOCITY 0x58
```

9.73.1.223 WZCNT

```
#define WZCNT 0x6d
```

9.73.1.224 WZSTATIC

```
#define WZSTATIC 0x6f
```

9.73.1.225 WZTIME

```
#define WZTIME 0x6e
```

9.73.1.226 XA30NSCALE

```
#define XA30NSCALE 0x8a
```

9.73.1.227 XA30PSCALE

```
#define XA30PSCALE 0x89
```

9.73.1.228 XA85NSCALE

```
#define XA85NSCALE 0x86
```

9.73.1.229 XA85PSCALE

```
#define XA85PSCALE 0x85
```

9.73.1.230 XMAX

```
#define XMAX 0x5b
```

9.73.1.231 XMIN

```
#define XMIN 0x5a
```

9.73.1.232 XREFROLL

```
#define XREFROLL 0x79
```

9.73.1.233 XRSCALE

```
#define XRSCALE 0x76
```

9.73.1.234 YA30NSCALE

```
#define YA30NSCALE 0x8c
```

9.73.1.235 YA30PSCALE

```
#define YA30PSCALE 0x8b
```

9.73.1.236 YA85NSCALE

```
#define YA85NSCALE 0x88
```

9.73.1.237 YA85PSCALE

```
#define YA85PSCALE 0x87
```

9.73.1.238 Yaw

```
#define Yaw 0x3f
```

9.73.1.239 YMAX

```
#define YMAX 0x5f
```

9.73.1.240 YMIN

```
#define YMIN 0x5e
```

9.73.1.241 YREFPITCH

```
#define YREFPITCH 0x7a
```

9.73.1.242 YRSCALE

```
#define YRSCALE 0x77
```

9.73.1.243 YYMM

```
#define YYMM 0x30
```

9.73.1.244 ZREFYAW

```
#define ZREFYAW 0x7b
```

9.73.1.245 ZRSCALE

```
#define ZRSCALE 0x78
```

9.74 REG.h

[Go to the documentation of this file.](#)

```
00001 #ifndef __AHRSREG_H
00002 #define __AHRSREG_H
00003
00004 #ifdef __cplusplus
00005 extern "C" {
00006 #endif
00007
00008 #define REGSIZE      0x90
00009
00010 #define SAVE         0x00
00011 #define CALSW        0x01
00012 #define RSW          0x02
00013 #define RRATE        0x03
00014 #define BAUD         0x04
00015 #define AXOFFSET     0x05
00016 #define AYOFFSET     0x06
00017 #define AZOFFSET     0x07
00018 #define GXOFFSET     0x08
00019 #define GYOFFSET     0x09
00020 #define GZOFFSET     0x0a
00021 #define HXOFFSET     0x0b
00022 #define HYOFFSET     0x0c
00023 #define HZOFFSET     0x0d
00024 #define D0MODE        0x0e
00025 #define D1MODE        0x0f
00026 #define D2MODE        0x10
00027 #define D3MODE        0x11
00028 #define D0PWMH       0x12
00029 #define D1PWMH       0x13
00030 #define D2PWMH       0x14
00031 #define D3PWMH       0x15
00032 #define D0PWMT      0x16
00033 #define D1PWMT      0x17
00034 #define D2PWMT      0x18
00035 #define D3PWMT      0x19
00036 #define IICADDR      0x1a
00037 #define LEDOFF       0x1b
00038 #define MAGRANGX    0x1c
00039 #define MAGRANGY    0x1d
00040 #define MAGRANGZ    0x1e
00041 #define BANDWIDTH   0x1f
00042 #define GYRORANGE   0x20
00043 #define ACCRANGE    0x21
00044 #define SLEEP        0x22
00045 #define ORIENT       0x23
00046 #define AXIS6       0x24
```

```
00047 #define FILTK          0x25
00048 #define GPSBAUD        0x26
00049 #define READADDR        0x27
00050 #define BWSCALE          0x28
00051 #define MOVETHR          0x28
00052 #define MOVESTA          0x29
00053 #define ACCFILT          0x2A
00054 #define GYROFILT          0x2b
00055 #define MAGFILT          0x2c
00056 #define POWONSEND        0x2d
00057 #define VERSION          0x2e
00058 #define CCBW             0x2f
00059 #define YYMM             0x30
00060 #define DDHH             0x31
00061 #define MMSS             0x32
00062 #define MS               0x33
00063 #define AX               0x34
00064 #define AY               0x35
00065 #define AZ               0x36
00066 #define GX               0x37
00067 #define GY               0x38
00068 #define GZ               0x39
00069 #define HXi              0x3a
00070 #define HYi              0x3b
00071 #define HZi              0x3c
00072 #define Roll             0x3d
00073 #define Pitch            0x3e
00074 #define Yaw               0x3f
00075 #define TEMP             0x40
00076
00077 /* High precision */
00078 #define LRoll            0x3d
00079 #define HRoll            0x3e
00080 #define LPitch            0x3f
00081 #define HPitch            0x40
00082 #define LYaw              0x41
00083 #define HYaw              0x42
00084 #define TEMP905x          0x43
00085
00086
00087 #define D0Status          0x41
00088 #define D1Status          0x42
00089 #define D2Status          0x43
00090 #define D3Status          0x44
00091 #define PressureL         0x45
00092 #define PressureH         0x46
00093 #define HeightL           0x47
00094 #define HeightH           0x48
00095 #define LonL              0x49
00096 #define LonH              0x4a
00097 #define LatL              0x4b
00098 #define LatH              0x4c
00099 #define GPSHeight         0x4d
00100
00101 #define GPSVX_L           0x4E
00102 #define GPSVX_H           0x4F
00103 #define GPSVY_L           0x50
00104 #define GPSVY_H           0x51
00105 #define GPSVZ_L           0x52
00106 #define GPSVZ_H           0x53
00107 #define GPSYAW            0x4e
00108
00109 #define GPSVL             0x4f
00110 #define GPSVH             0x50
00111 #define q0                0x51
00112 #define q1                0x52
00113 #define q2                0x53
00114 #define q3                0x54
00115 #define SVNUM            0x55
00116 #define PDOP              0x56
00117 #define HDOP              0x57
00118 #define VDOP              0x58
00119 #define DELAYT            0x59
00120 #define XMIN              0x5a
00121 #define XMAX              0x5b
00122 #define BATVAL            0x5c
00123 #define ALARMPIN          0x5d
00124 #define YMINT             0x5e
00125 #define YMAX              0x5f
00126 #define GYROZSCALE          0x60
00127 #define GYROCALITHR         0x61
00128 #define ALARMLEVEL          0x62
00129 #define GYROCALTIME         0x63
00130 #define REFROLL            0x64
00131 #define REFPITCH            0x65
00132 #define REFYAW              0x66
00133 #define GPSTYPE            0x67
```

```

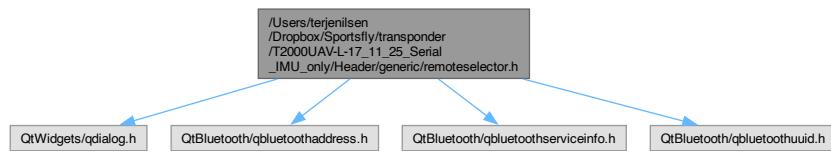
00134 #define TRIGTIME      0x68
00135 #define KEY          0x69
00136 #define WERROR        0x6a
00137 #define TIMEZONE      0x6b
00138 #define CALICNT       0x6c
00139 #define WZCNT          0x6d
00140 #define WZTIME         0x6e
00141 #define WZSTATIC        0x6f
00142 #define ACCSENSOR      0x70
00143 #define GYROSENSOR     0x71
00144 #define MAGSENSOR      0x72
00145 #define PRESSENSOR     0x73
00146 #define MODDELAY        0x74
00147
00148 #define ANGLEAXIS      0x75
00149 #define XRSCALE         0x76
00150 #define YRSCALE         0x77
00151 #define ZRSCALE         0x78
00152
00153 #define XREFROLL        0x79
00154 #define YREFPITCH       0x7a
00155 #define ZREFYAW         0x7b
00156
00157 #define ANGXOFFSET      0x7c
00158 #define ANGYOFFSET      0x7d
00159 #define ANGZOFFSET      0x7e
00160
00161 #define NUMBERID1       0x7f
00162 #define NUMBERID2       0x80
00163 #define NUMBERID3       0x81
00164 #define NUMBERID4       0x82
00165 #define NUMBERID5       0x83
00166 #define NUMBERID6       0x84
00167
00168 #define XA85PSCALE      0x85
00169 #define XA85NSCALE      0x86
00170 #define YA85PSCALE      0x87
00171 #define YA85NSCALE      0x88
00172 #define XA30PSCALE      0x89
00173 #define XA30NSCALE      0x8a
00174 #define YA30PSCALE      0x8b
00175 #define YA30NSCALE      0x8c
00176
00177 #define CHIPIDL         0x8D
00178 #define CHIPIDH         0x8E
00179 #define REGINITFLAG     REGSIZE-1
00180
00181
00182 /* AXIS6 */
00183 #define ALGRITHM9       0
00184 #define ALGRITHM6       1
00185
00186 /*****CALSW*****/
00187 #define NORMAL          0x00
00188 #define CALGYROACC      0x01
00189 #define CALMAG          0x02
00190 #define CALALTITUDE     0x03
00191 #define CALANGLEZ       0x04
00192 #define CALACCL         0x05
00193 #define CALACCR         0x06
00194 #define CALMAGMM        0x07
00195 #define CALREFANGLE     0x08
00196 #define CALMAG2STEP      0x09
00197 // #define CALACCX        0x09
00198 // #define ACC45PRX       0x0A
00199 // #define ACC45NRX       0x0B
00200 // #define CALACCY        0x0C
00201 // #define ACC45PRY       0x0D
00202 // #define ACC45NRY       0x0E
00203 // #define CALREFANGLER    0x0F
00204 // #define CALACCINIT     0x10
00205 // #define CALREFANGLEINIT 0x11
00206 #define CALHEXAHEDRON    0x12
00207
00208 /*****OUTPUTHEAD*****/
00209 #define WIT_TIME         0x50 // OK
00210 #define WIT_ACC          0x51 // OK OK
00211 #define WIT_GYRO         0x52 // OK OK
00212 #define WIT_ANGLE        0x53 // OK OK
00213 #define WIT_MAGNETIC     0x54 //
00214 #define WIT_DPORT        0x55 // OK OK
00215 #define WIT_PRESS        0x56 //
00216 #define WIT_GPS          0x57 // OK OK
00217 #define WIT_VELOCITY     0x58 // OK
00218 #define WIT_QUATER       0x59 //
00219 #define WIT_GSA          0x5A //
00220 #define WIT_REGVALUE      0x5F // OK OK

```

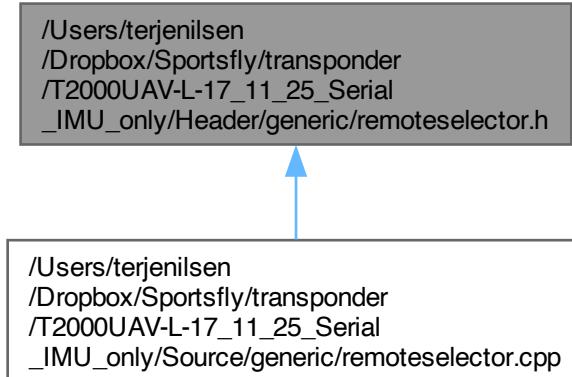
```
00221
00222
00223 /******RSW******/
00224 #define RSW_TIME          0x01
00225 #define RSW_ACC           0x02
00226 #define RSW_GYRO          0x04
00227 #define RSW_ANGLE          0x08
00228 #define RSW_MAG            0x10
00229 #define RSW_PORT           0x20
00230 #define RSW_PRESS          0x40
00231 #define RSW_GPS            0x80
00232 #define RSW_V              0x100
00233 #define RSW_Q              0x200
00234 #define RSW_GSA            0x400
00235 #define RSW_MASK           0xffff
00236
00238 #define RRATE_NONE         0x0d
00239 #define RRATE_02HZ         0x01
00240 #define RRATE_05HZ         0x02
00241 #define RRATE_1HZ          0x03
00242 #define RRATE_2HZ          0x04
00243 #define RRATE_5HZ          0x05
00244 #define RRATE_10HZ         0x06
00245 #define RRATE_20HZ         0x07
00246 #define RRATE_50HZ         0x08
00247 #define RRATE_100HZ        0x09
00248 #define RRATE_125HZ        0x0a //only WT931
00249 #define RRATE_200HZ        0x0b
00250 #define RRATE_ONCE          0x0c
00251
00252 /* BAUD */
00253 #define WIT_BAUD_4800       1
00254 #define WIT_BAUD_9600       2
00255 #define WIT_BAUD_19200      3
00256 #define WIT_BAUD_38400      4
00257 #define WIT_BAUD_57600      5
00258 #define WIT_BAUD_115200     6
00259 #define WIT_BAUD_230400     7
00260 #define WIT_BAUD_460800     8
00261 #define WIT_BAUD_921600     9
00262
00263 /*CAN BAUD*/
00264 #define CAN_BAUD_1000000    0
00265 #define CAN_BAUD_800000     1
00266 #define CAN_BAUD_500000     2
00267 #define CAN_BAUD_400000     3
00268 #define CAN_BAUD_250000     4
00269 #define CAN_BAUD_200000     5
00270 #define CAN_BAUD_125000     6
00271 #define CAN_BAUD_100000     7
00272 #define CAN_BAUD_80000      8
00273 #define CAN_BAUD_50000      9
00274 #define CAN_BAUD_40000     10
00275 #define CAN_BAUD_20000     11
00276 #define CAN_BAUD_10000     12
00277 #define CAN_BAUD_5000      13
00278 #define CAN_BAUD_3000      14
00279
00280 /* KEY */
00281 #define KEY_UNLOCK          0xB588
00282
00283 /* SAVE */
00284 #define SAVE_PARAM          0x00
00285 #define SAVE_SWRST          0xFF
00286
00287 /* ORIENT */
00288 #define ORIENT_HERIZONE      0
00289 #define ORIENT_VERTICLE      1
00290
00291 /* BANDWIDTH */
00292 #define BANDWIDTH_256HZ      0
00293 #define BANDWIDTH_184HZ      1
00294 #define BANDWIDTH_94HZ       2
00295 #define BANDWIDTH_44HZ       3
00296 #define BANDWIDTH_21HZ       4
00297 #define BANDWIDTH_10HZ       5
00298 #define BANDWIDTH_5HZ        6
00299
00300
00301 #ifdef __cplusplus
00302 }
00303 #endif
00304
00305 #endif
```

9.75 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header/generic/remoteselector.h File Reference

```
#include <QtWidgets/qdialog.h>
#include <QtBluetooth/qbluetoothaddress.h>
#include <QtBluetooth/qbluetoothserviceinfo.h>
#include <QtBluetooth/qbluetoothuuid.h>
Include dependency graph for remoteselector.h:
```



This graph shows which files directly or indirectly include this file:



Data Structures

- class [RemoteSelector](#)

Namespaces

- namespace [Ui](#)

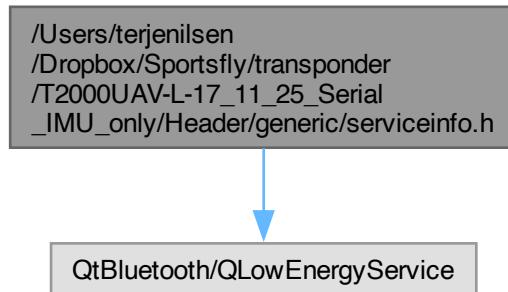
9.76 remoteselector.h

[Go to the documentation of this file.](#)

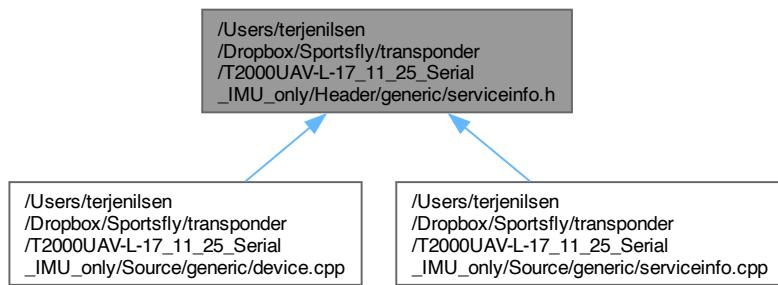
```
00001 // Copyright (C) 2017 The Qt Company Ltd.
00002 // SPDX-License-Identifier: LicenseRef-Qt-Commercial OR BSD-3-Clause
00003
00004 #ifndef REMOTESELECTOR_H
00005 #define REMOTESELECTOR_H
00006
00007 #include <QtWidgets/qdialog.h>
00008
00009 #include <QtBluetooth/qbluetoothaddress.h>
00010 #include <QtBluetooth/qbluetoothserviceinfo.h>
00011 #include <QtBluetooth/qbluetoothuuid.h>
00012
00013 QT_FORWARD_DECLARE_CLASS(QBluetoothServiceDiscoveryAgent)
00014 QT_FORWARD_DECLARE_CLASS(QLWidgetItem)
00015
00016 QT_USE_NAMESPACE
00017
00018 QT_BEGIN_NAMESPACE
00019 namespace Ui {
00020     class RemoteSelector;
00021 }
00022 QT_END_NAMESPACE
00023
00024 class RemoteSelector : public QDialog
00025 {
00026     Q_OBJECT
00027
00028 public:
00029     explicit RemoteSelector(const QBluetoothAddress &localAdapter, QWidget *parent = nullptr);
00030     ~RemoteSelector();
00031
00032     void startDiscovery(const QBluetoothUuid &uuid);
00033     void stopDiscovery();
00034     QBluetoothServiceInfo service() const;
00035
00036 private:
00037     Ui::RemoteSelector *ui;
00038
00039     QBluetoothServiceDiscoveryAgent *m_discoveryAgent;
00040     QBluetoothServiceInfo m_service;
00041     QMap<QLWidgetItem *, QBluetoothServiceInfo> m_discoveredServices;
00042
00043 private slots:
00044     void serviceDiscovered(const QBluetoothServiceInfo &serviceInfo);
00045     void discoveryFinished();
00046     void on_remoteDevices_itemActivated(QLWidgetItem *item);
00047     void on_remoteDevices_itemClicked(QLWidgetItem *item);
00048     void on_cancelButton_clicked();
00049     void on_connectButton_clicked();
00050 };
00051
00052 #endif // REMOTESELECTOR_H
```

9.77 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Header/generic/serviceinfo.h File Reference

```
#include <QtBluetooth/QLowEnergyService>
Include dependency graph for serviceinfo.h:
```



This graph shows which files directly or indirectly include this file:



Data Structures

- class [ServiceInfo](#)

9.78 serviceinfo.h

[Go to the documentation of this file.](#)

```
00001 /*****
00002 /**
00003 ** Copyright (C) 2013 BlackBerry Limited. All rights reserved.
00004 ** Copyright (C) 2017 The Qt Company Ltd.
00005 ** Contact: https://www.qt.io/licensing/
00006 **/
```

```

00007 ** This file is part of the QtBluetooth module of the Qt Toolkit.
00008 **
00009 ** $QT_BEGIN_LICENSE:BSD$
00010 ** Commercial License Usage
00011 ** Licensees holding valid commercial Qt licenses may use this file in
00012 ** accordance with the commercial license agreement provided with the
00013 ** Software or, alternatively, in accordance with the terms contained in
00014 ** a written agreement between you and The Qt Company. For licensing terms
00015 ** and conditions see https://www.qt.io/terms-conditions. For further
00016 ** information use the contact form at https://www.qt.io/contact-us.
00017 **
00018 ** BSD License Usage
00019 ** Alternatively, you may use this file under the terms of the BSD license
00020 ** as follows:
00021 **
00022 ** Redistributions and use in source and binary forms, with or without
00023 ** modification, are permitted provided that the following conditions are
00024 ** met:
00025 **      * Redistributions of source code must retain the above copyright
00026 ** notice, this list of conditions and the following disclaimer.
00027 **      * Redistributions in binary form must reproduce the above copyright
00028 ** notice, this list of conditions and the following disclaimer in
00029 ** the documentation and/or other materials provided with the
00030 ** distribution.
00031 **      * Neither the name of The Qt Company Ltd nor the names of its
00032 ** contributors may be used to endorse or promote products derived
00033 ** from this software without specific prior written permission.
00034 **
00035 **
00036 ** THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
00037 ** "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
00038 ** LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
00039 ** A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
00040 ** OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
00041 ** SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
00042 ** LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
00043 ** DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
00044 ** THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
00045 ** (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
00046 ** OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE."
00047 **
00048 ** $QT_END_LICENSE$
00049 **
00050 ****
00051
00052 #ifndef SERVICEINFO_H
00053 #define SERVICEINFO_H
00054 #include <QtBluetooth/QLowEnergyService>
00055
00056 class ServiceInfo: public QObject
00057 {
00058     Q_OBJECT
00059
00060 public:
00061     ServiceInfo() = default;
00062     ServiceInfo(QLowEnergyService *service);
00063     QLowEnergyService *service() const;
00064     QString getUuid() const;
00065     QString getName() const;
00066     QString getType() const;
00067
00068 Q_SIGNALS:
00069     void serviceChanged();
00070
00071 private:
00072     QLowEnergyService *m_service = nullptr;
00073 };
00074
00075 #endif // SERVICEINFO_H

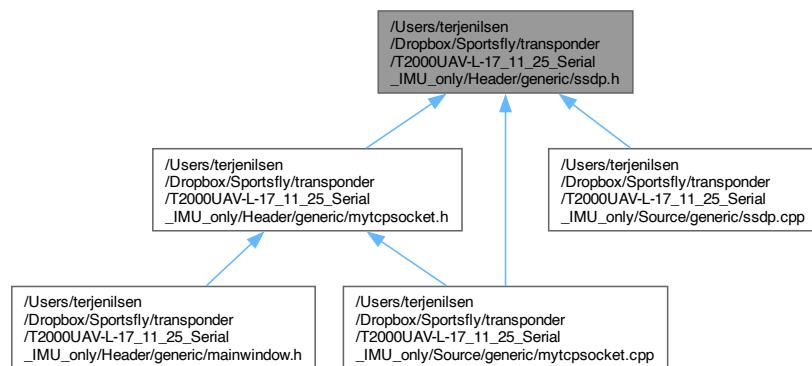
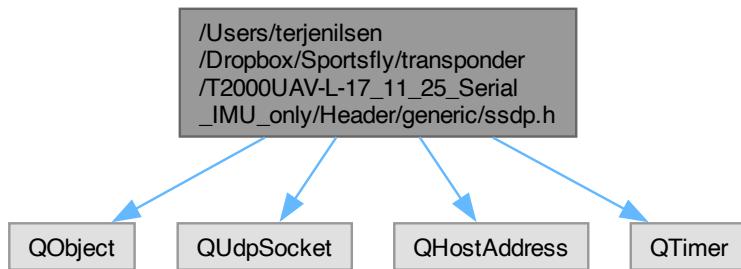
```

9.79 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_← 11_25_Serial_IMU_only/Header/generic/ssdp.h File Reference

SSDP (Simple Service Discovery Protocol) discovery support.

```
#include <QObject>
#include <QUdpSocket>
#include <QHostAddress>
```

```
#include <QTimer>
Include dependency graph for ssdp.h:
```



Data Structures

- struct [SensorService](#)
- class [SsdpDiscoverer](#)

Macros

- #define [SENSOR "IMU"](#)
- #define [SENSOR_TYPE "Airplane-device"](#)

Functions

- static const QHostAddress [SSDP_GROUP \("255.255.255.255"\)](#)

Variables

- static const quint16 `SSDP_PORT` = 4210
- static const char * `SSDP_ST` = `SENSOR_TYPE`

9.79.1 Detailed Description

SSDP (Simple Service Discovery Protocol) discovery support.

Declares the `SsdpDiscoverer` class, which provides basic SSDP discovery functionality using UDP multicast and integrates with Qt's event loop.

This header is part of a larger Qt 6 project. Public API stability is required; names and signatures must not change.

9.79.2 Macro Definition Documentation

9.79.2.1 SENSOR

```
#define SENSOR "IMU"
```

9.79.2.2 SENSOR_TYPE

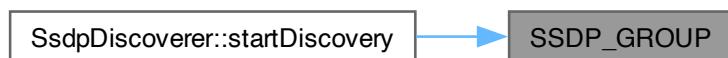
```
#define SENSOR_TYPE "Airplane-device"
```

9.79.3 Function Documentation

9.79.3.1 SSDP_GROUP()

```
const QHostAddress SSDP_GROUP (
    "255.255.255.255" ) [static]
```

Here is the caller graph for this function:



9.79.4 Variable Documentation

9.79.4.1 SSDP_PORT

```
const quint16 SSDP_PORT = 4210 [static]
```

9.79.4.2 SSDP_ST

```
const char* SSDP_ST = SENSOR_TYPE [static]
```

9.80 ssdp.h

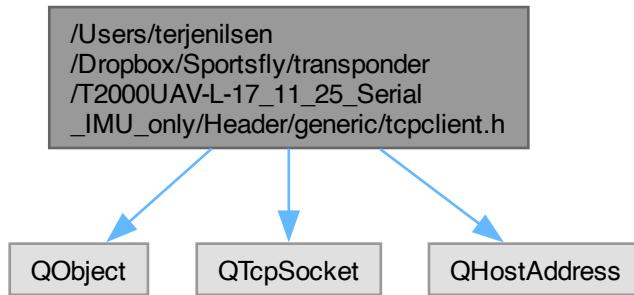
[Go to the documentation of this file.](#)

```
00001 // SsdpDiscoverer.h
00002 #pragma once
00003
00014
00015 #include <QObject>
00016 #include <QUdpSocket>
00017 #include <QHostAddress>
00018 #include < QTimer>
00019
00020 #define SENSOR      "IMU"           // IMU, RADAR, TRANSPONDER, etc.
00021 #define SENSOR_TYPE "Airplane-device" // change per firmware type
00022
00023 static const QHostAddress SSDP_GROUP("255.255.255.255");
00024 static const quint16 SSDP_PORT = 4210;
00025
00026 struct SensorService {
00027     const char *name;    // logical name, e.g. "IMU"
00028     const char *st;      // ST value, e.g. "imu-device"
00029     const char *uuid;    // per-service UUID
00030 };
00031
00032 // Our custom service type
00033 static const char *SSDP_ST = SENSOR_TYPE;
00034
00035 class SsdpDiscoverer : public QObject
00036 {
00037     Q_OBJECT
00038 public:
00039     explicit SsdpDiscoverer(QObject *parent = nullptr);
00040
00041     // Start a discovery burst (one M-SEARCH + listen for replies for a few seconds)
00042     void startDiscovery(int durationMs = 2000);
00043
00044 signals:
00045     void deviceFound(const QHostAddress &addr,
00046                      quint16 port,
00047                      const QString &st);
00048
00049 private slots:
00050     void onReadyRead();
00051
00052 private:
00053     QUdpSocket m_socket;
00054     QTimer     m_timer;
00055 };
```

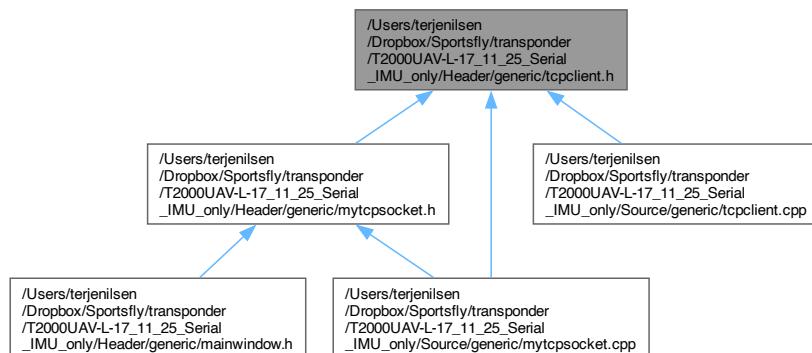
9.81 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_← 11_25_Serial_IMU_only/Header/generic/tcpclient.h File Reference

```
#include <QObject>
#include <QTcpSocket>
#include <QHostAddress>
```

Include dependency graph for tcpclient.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- class [TcpClient](#)

9.82 tcpclient.h

[Go to the documentation of this file.](#)

```

00001 #pragma once
00002
00003 #include <QObject>
00004 #include <QTcpSocket>
00005 #include <QHostAddress>
00006
00007
00008 class TcpClient : public QObject
00009 {
0010     Q_OBJECT
0011 public:
0012     explicit TcpClient(QObject *parent = nullptr);
  
```

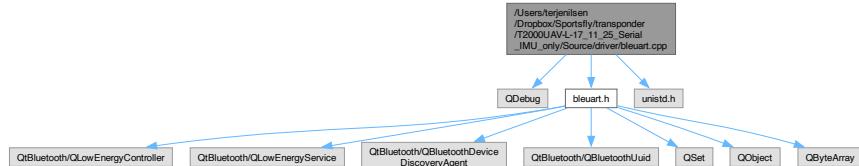
```

00013
00014     // Call this with the ESP32 IP and port (23 in your case)
00015     void connectTo(const QHostAddress &address, quint16 port = 23);
00016     void disconnectFrom();
00017
00018     // Sends raw bytes (for telnet-style text, you can append "\r\n")
00019     void sendData(const QByteArray &data);
00020
00021 signals:
00022     void connected();
00023     void disconnected();
00024     void errorOccurred(const QString &error);
00025     void dataReceived(const QByteArray &data);
00026
00027 private slots:
00028     void onConnected();
00029     void onDisconnected();
00030     void onReadyRead();
00031     void onErrorOccurred(QAbstractSocket::SocketError socketError);
00032
00033 private:
00034     QTcpSocket m_socket;
00035 };

```

9.83 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/driver/bleuart.cpp File Reference

```
#include <QDebug>
#include "bleuart.h"
#include <unistd.h>
Include dependency graph for bleuart.cpp:
```

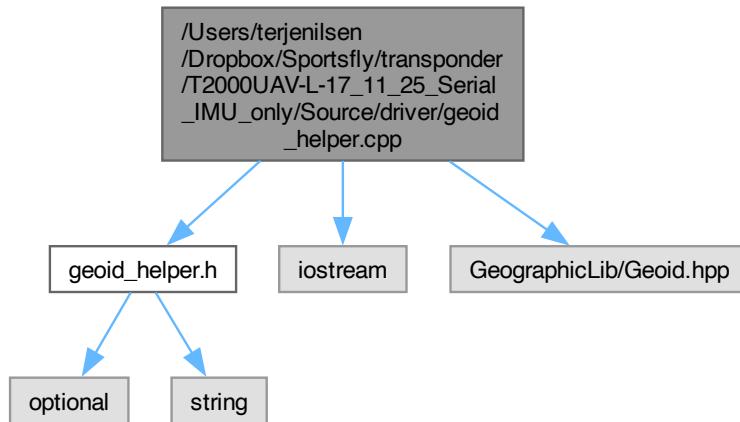


9.84 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/driver/geoid_helper.cpp File Reference

Geoid helper implementation.

```
#include "geoid_helper.h"
#include <iostream>
#include <GeographicLib/Geoid.hpp>
```

Include dependency graph for geoid_helper.cpp:



Data Structures

- struct [GeoidHelper::Impl](#)

9.84.1 Detailed Description

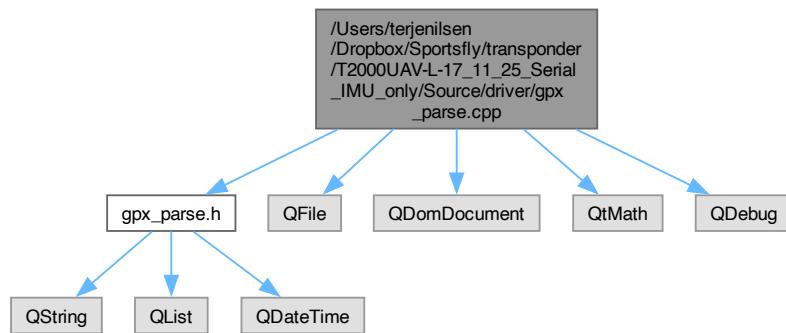
Geoid helper implementation.

Contains the private implementation details for geoid calculations.

9.85 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/driver/gpx_parse.cpp File Reference

```
#include "gpx_parse.h"
#include <QFile>
#include <QDomDocument>
#include <QtMath>
```

```
#include <QDebug>
Include dependency graph for gpx_parse.cpp:
```



Functions

- double [haversineDistance](#) (double lat1, double lon1, double lat2, double lon2)

Variables

- static constexpr double [EARTH_RADIUS](#) = 6371000.0

9.85.1 Function Documentation

9.85.1.1 haversineDistance()

```
double haversineDistance (
    double lat1,
    double lon1,
    double lat2,
    double lon2)
```

Here is the caller graph for this function:



9.85.2 Variable Documentation

9.85.2.1 EARTH_RADIUS

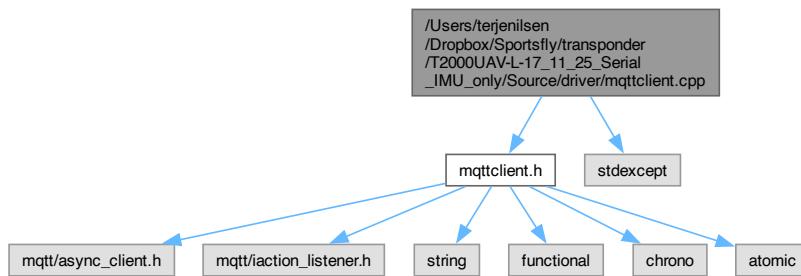
```
double EARTH_RADIUS = 6371000.0 [static], [constexpr]
```

9.86 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/driver/mqttclient.cpp File Reference

Implementation of [MqttClient](#).

```
#include "mqttclient.h"
#include <stdexcept>
```

Include dependency graph for mqttclient.cpp:



9.86.1 Detailed Description

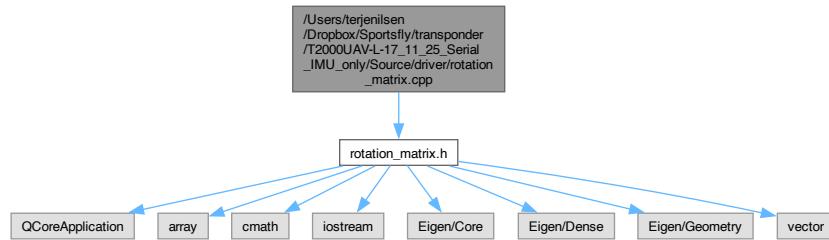
Implementation of [MqttClient](#).

Contains the implementation details for the [MqttClient](#) class.

9.87 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/driver/rotation_matrix.cpp File Reference

Implementation of createRotationMatrix.

```
#include "rotation_matrix.h"
Include dependency graph for rotation_matrix.cpp:
```



Functions

- `Matrix3x3 createRotationMatrixZ (double theta)`
- `Matrix3x3 createRotationMatrix (Vector3x theta)`
- `Matrix3x3 multiplyMatrix (const Matrix3x3 &mat1, const Matrix3x3 &mat2)`
- `Vector3x rotateVector (const Matrix3x3 &rotationMatrix, const Vector3x &vector)`
- `void rotateSensors (Vector3x &gyro, Vector3x &accel, Vector3x &mag, Vector3x theta)`
- `void rotateSensors (Vector3x &gyro, Vector3x &accel, Vector3x &mag, Vector3x &attitude, Matrix3x3 rotationMatrix)`
- `void rotateSensors (Vector3x &gyro, Vector3x &accel, Vector3x &mag, Matrix3x3 rotationMatrix)`
- `void printVector (const Vector3x &vector)`

9.87.1 Detailed Description

Implementation of `createRotationMatrix`.

Contains the implementation details for the `createRotationMatrix` class.

9.87.2 Function Documentation

9.87.2.1 `createRotationMatrix()`

```
Matrix3x3 createRotationMatrix (
    Vector3x theta)
```

Here is the call graph for this function:



Here is the caller graph for this function:



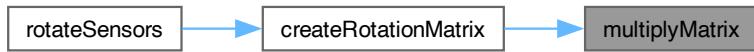
9.87.2.2 createRotationMatrixZ()

```
Matrix3x3 createRotationMatrixZ (
    double theta)
```

9.87.2.3 multiplyMatrix()

```
Matrix3x3 multiplyMatrix (
    const Matrix3x3 & mat1,
    const Matrix3x3 & mat2)
```

Here is the caller graph for this function:



9.87.2.4 printVector()

```
void printVector (
    const Vector3x & vector)
```

9.87.2.5 rotateSensors() [1/3]

```
void rotateSensors (
    Vector3x & gyro,
    Vector3x & accel,
    Vector3x & mag,
    Matrix3x3 rotationMatrix)
```

Here is the call graph for this function:



9.87.2.6 `rotateSensors()` [2/3]

```
void rotateSensors (
    Vector3x & gyro,
    Vector3x & accel,
    Vector3x & mag,
    Vector3x & attitude,
    Matrix3x3 rotationMatrix)
```

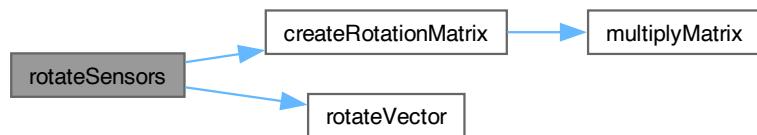
Here is the call graph for this function:



9.87.2.7 `rotateSensors()` [3/3]

```
void rotateSensors (
    Vector3x & gyro,
    Vector3x & accel,
    Vector3x & mag,
    Vector3x theta)
```

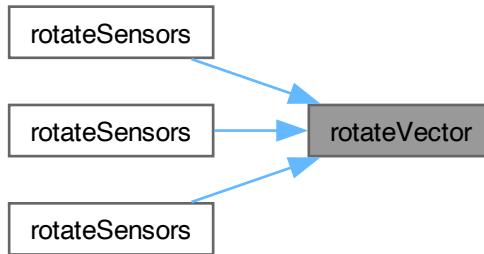
Here is the call graph for this function:



9.87.2.8 rotateVector()

```
Vector3x rotateVector (
    const Matrix3x3 & rotationMatrix,
    const Vector3x & vector)
```

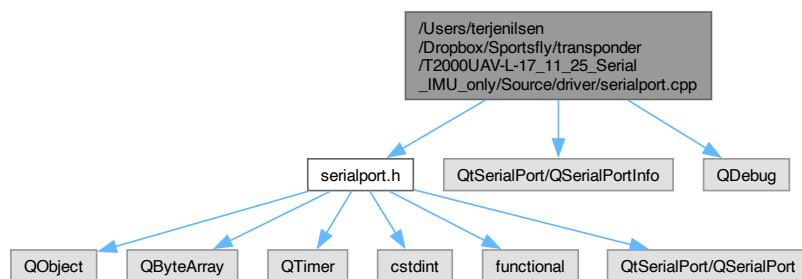
Here is the caller graph for this function:



9.88 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/driver/serialport.cpp File Reference

Implementation of [ComQt](#).

```
#include "serialport.h"
#include <QtSerialPort/QSerialPortInfo>
#include <QDebug>
Include dependency graph for serialport.cpp:
```



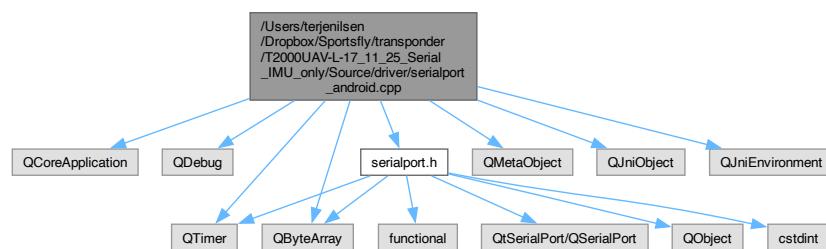
9.88.1 Detailed Description

Implementation of [ComQt](#).

Contains the implementation details for the [ComQt](#) class.

9.89 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/driver/serialport_android.cpp File Reference

```
#include <QCoreApplication>
#include <QDebug>
#include <QTimer>
#include <QByteArray>
#include <QMetaObject>
#include <QJniObject>
#include <QJniEnvironment>
#include "serialport.h"
Include dependency graph for serialport_android.cpp:
```



Functions

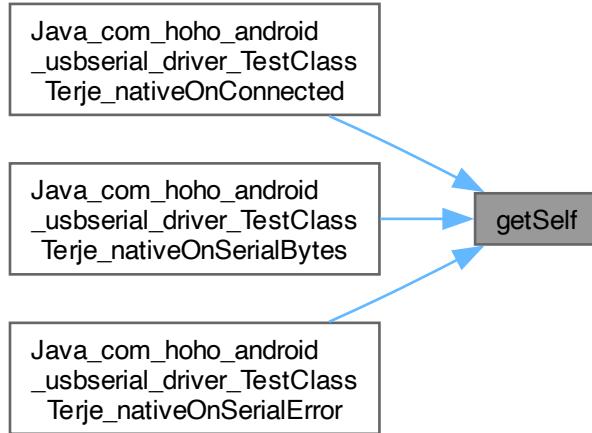
- static ComQt * [getSelf](#) (JNIEnv *env, jobject thiz)
- JNIEXPORT void JNICALL [Java_com_hoho_android_usbserial_driver_TestClassTerje_nativeOnSerialBytes](#) (JNIEnv *env, jobject thiz, jbyteArray jdata)
- JNIEXPORT void JNICALL [Java_com_hoho_android_usbserial_driver_TestClassTerje_nativeOnConnected](#) (JNIEnv *env, jobject thiz, jboolean jconnected)
- JNIEXPORT void JNICALL [Java_com_hoho_android_usbserial_driver_TestClassTerje_nativeOnSerialError](#) (JNIEnv *env, jobject thiz, jstring jmsg)

9.89.1 Function Documentation

9.89.1.1 [getSelf\(\)](#)

```
ComQt * getSelf (
    JNIEnv * env,
    jobject thiz) [static]
```

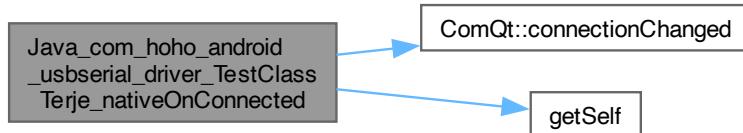
Here is the caller graph for this function:



9.89.1.2 Java_com_hoho_android_usbserial_driver_TestClassTerje_nativeOnConnected()

```
JNIEXPORT void JNICALL Java_com_hoho_android_usbserial_driver_TestClassTerje_nativeOnConnected
(
    JNIEnv * env,
    jobject thiz,
    jboolean jconnected)
```

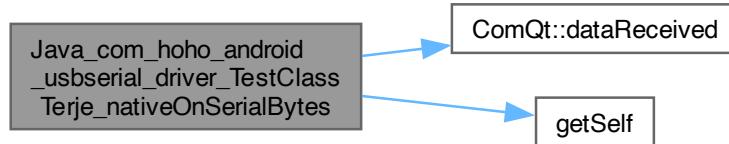
Here is the call graph for this function:



9.89.1.3 Java_com_hoho_android_usbserial_driver_TestClassTerje_nativeOnSerialBytes()

```
JNIEXPORT void JNICALL Java_com_hoho_android_usbserial_driver_TestClassTerje_nativeOnSerialBytes (
    JNIEnv * env,
    jobject thiz,
    jbyteArray jdata)
```

Here is the call graph for this function:

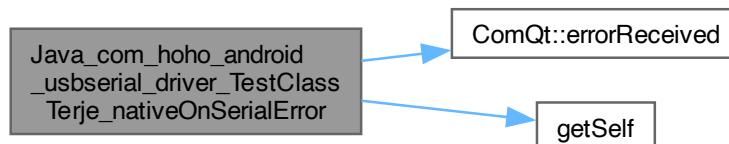


9.89.1.4 Java_com_hoho_android_usbserial_driver_TestClassTerje_nativeOnSerialError()

```

JNIEXPORT void JNICALL Java_com_hoho_android_usbserial_driver_TestClassTerje_nativeOnSerialError (
    JNIEnv * env,
    jobject thiz,
    jstring jmsg)
  
```

Here is the call graph for this function:

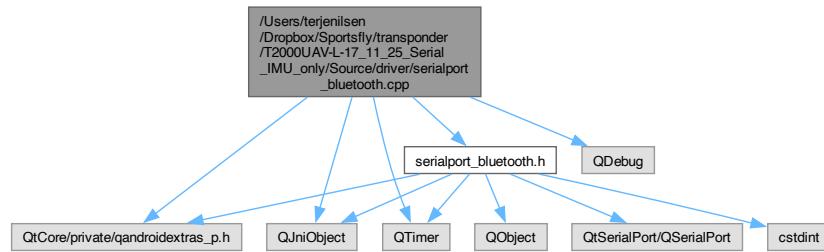


9.90 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/driver/serialport_bluetooth.cpp File Reference

```

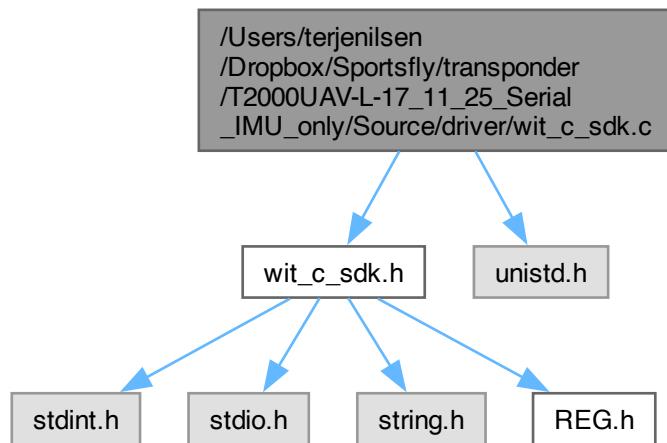
#include <QtCore/private/qandroidextras_p.h>
#include <QJniObject>
#include <QDebug>
#include <QTimer>
  
```

```
#include "serialport_bluetooth.h"  
Include dependency graph for serialport_bluetooth.cpp:
```



9.91 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/driver/wit_c_sdk.c File Reference

```
#include "wit_c_sdk.h"  
#include <unistd.h>  
Include dependency graph for wit_c_sdk.c:
```



Macros

- #define FuncW 0x06
- #define FuncR 0x03

Functions

- static uint16_t **`__CRC16`** (uint8_t *puchMsg, uint16_t usDataLen)
- static uint8_t **`__CaliSum`** (uint8_t *data, uint32_t len)
- int32_t **`WitSerialWriteRegister`** (**`SerialWrite`** Write_func)
- static void **`CopeWitData`** (uint8_t ucIndex, uint16_t *p_data, uint32_t uiLen)
- void **`WitSerialDataIn`** (void *x, const char *ucData, uint32_t length)
- int32_t **`WitRegisterCallBack`** (**`RegUpdateCb`** update_func)
- int32_t **`WitWriteReg`** (uint32_t uiReg, uint16_t usData)
- int32_t **`WitReadReg`** (uint32_t uiReg, uint32_t uiReadNum)
- int32_t **`WitInit`** (uint32_t uiProtocol, uint8_t ucAddr)
- void **`WitDeInit`** (void)
- void **`posix_delay_ms`** (uint16_t ms)
- int32_t **`WitDelayMsRegister`** (**`DelaymsCb`** delayms_func)
- char **`CheckRange`** (short sTemp, short sMin, short sMax)
- int32_t **`WitStartAccCali`** (void)
- int32_t **`WitStopAccCali`** (void)
- int32_t **`WitStartMagCali`** (void)
- int32_t **`WitStopMagCali`** (void)
- int32_t **`WitSetUartBaud`** (int32_t uiBaudIndex)
- int32_t **`WitSetCanBaud`** (int32_t uiBaudIndex)
- int32_t **`WitSetBandwidth`** (int32_t uiBaudWidth)
- int32_t **`WitSetOutputRate`** (int32_t uiRate)
- int32_t **`WitSetContent`** (int32_t uiRsw)
- int32_t **`WitSaveParameter`** ()
- int32_t **`WitSetForReset`** ()
- int32_t **`WitCaliRefAngle`** (void)

Variables

- static **`SerialWrite`** p_WitSerialWriteFunc = NULL
- static **`RegUpdateCb`** p_WitRegUpdateCbFunc = NULL
- static **`DelaymsCb`** p_WitDelaymsFunc = NULL
- static uint8_t **`s_ucAddr`** = 0xff
- static uint8_t **`s_ucWitDataBuff`** [WIT_DATA_BUFF_SIZE]
- static uint32_t **`s_uiWitDataCnt`** = 0
- static uint32_t **`s_uiProtocol`** = 0
- static uint32_t **`s_uiReadRegIndex`** = 0
- int16_t **`sReg`** [REGSIZE]
- static const uint8_t **`__ auchCRCHi`** [256]
- static const uint8_t **`__ auchCRCLo`** [256]

9.91.1 Macro Definition Documentation

9.91.1.1 FuncR

```
#define FuncR 0x03
```

9.91.1.2 FuncW

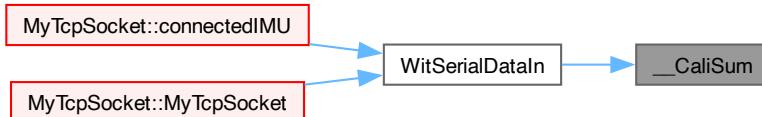
```
#define FuncW 0x06
```

9.91.2 Function Documentation

9.91.2.1 __CaliSum()

```
uint8_t __CaliSum (
    uint8_t * data,
    uint32_t len) [static]
```

Here is the caller graph for this function:



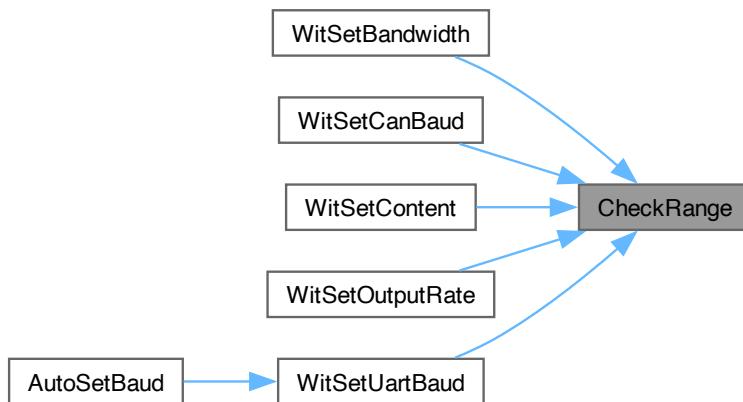
9.91.2.2 __CRC16()

```
uint16_t __CRC16 (
    uint8_t * puchMsg,
    uint16_t usDataLen) [static]
```

9.91.2.3 CheckRange()

```
char CheckRange (
    short sTemp,
    short sMin,
    short sMax)
```

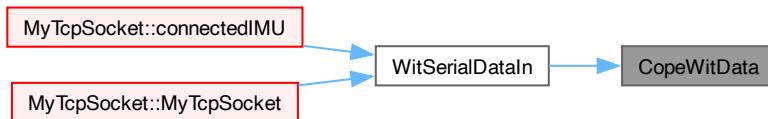
Here is the caller graph for this function:



9.91.2.4 CopeWitData()

```
void CopeWitData (
    uint8_t ucIndex,
    uint16_t * p_data,
    uint32_t uiLen) [static]
```

Here is the caller graph for this function:



9.91.2.5 posix_delay_ms()

```
void posix_delay_ms (
    uint16_t ms)
```

Here is the caller graph for this function:



9.91.2.6 WitCaliRefAngle()

```
int32_t WitCaliRefAngle (
    void )
```

Here is the call graph for this function:



9.91.2.7 WitDeInit()

```
void WitDeInit (
    void )
```

9.91.2.8 WitDelayMsRegister()

```
int32_t WitDelayMsRegister (
    DelaymsCb delayms_func)
```

Here is the caller graph for this function:



9.91.2.9 WitInit()

```
int32_t WitInit (
    uint32_t uiProtocol,
    uint8_t ucAddr)
```

Here is the caller graph for this function:



9.91.2.10 WitReadReg()

```
int32_t WitReadReg (
    uint32_t uiReg,
    uint32_t uiReadNum)
```

Here is the caller graph for this function:



9.91.2.11 WitRegisterCallBack()

```
int32_t WitRegisterCallBack (
    RegUpdateCb update_func)
```

Here is the caller graph for this function:



9.91.2.12 WitSaveParameter()

```
int32_t WitSaveParameter (
    void )
```

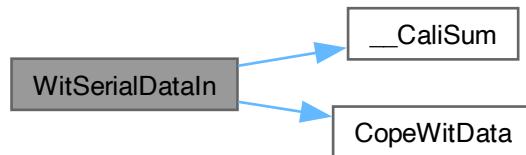
Here is the call graph for this function:



9.91.2.13 WitSerialDataIn()

```
void WitSerialDataIn (
    void * x,
    const char * ucData,
    uint32_t length)
```

Here is the call graph for this function:



Here is the caller graph for this function:



9.91.2.14 WitSerialWriteRegister()

```
int32_t WitSerialWriteRegister (
    SerialWrite Write_func)
```

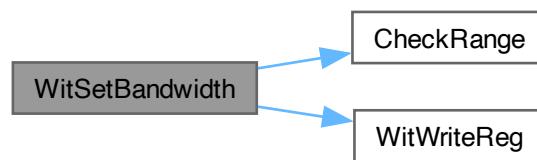
Here is the caller graph for this function:



9.91.2.15 WitSetBandwidth()

```
int32_t WitSetBandwidth (
    int32_t uiBaudWidth)
```

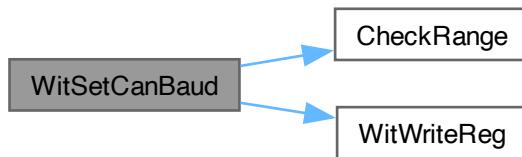
Here is the call graph for this function:



9.91.2.16 WitSetCanBaud()

```
int32_t WitSetCanBaud (
    int32_t uiBaudIndex)
```

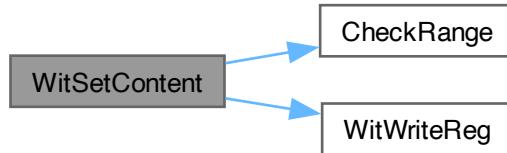
Here is the call graph for this function:



9.91.2.17 WitSetContent()

```
int32_t WitSetContent (
    int32_t uiRsw)
```

Here is the call graph for this function:



9.91.2.18 WitSetForReset()

```
int32_t WitSetForReset (
    void )
```

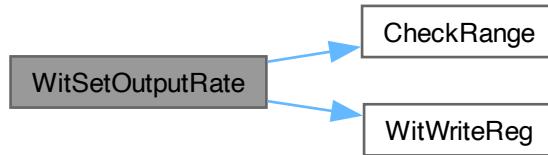
Here is the call graph for this function:



9.91.2.19 WitSetOutputRate()

```
int32_t WitSetOutputRate (
    int32_t uiRate)
```

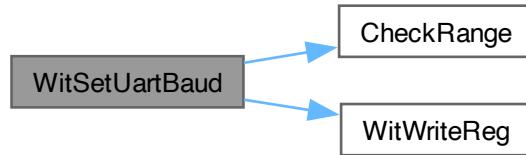
Here is the call graph for this function:



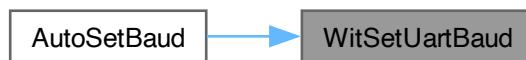
9.91.2.20 WitSetUartBaud()

```
int32_t WitSetUartBaud (
    int32_t uiBaudIndex)
```

Here is the call graph for this function:



Here is the caller graph for this function:



9.91.2.21 WitStartAccCali()

```
int32_t WitStartAccCali (
    void )
```

Here is the call graph for this function:



9.91.2.22 WitStartMagCali()

```
int32_t WitStartMagCali (
    void )
```

Here is the call graph for this function:



9.91.2.23 WitStopAccCali()

```
int32_t WitStopAccCali (
    void )
```

Here is the call graph for this function:



9.91.2.24 WitStopMagCali()

```
int32_t WitStopMagCali (
    void )
```

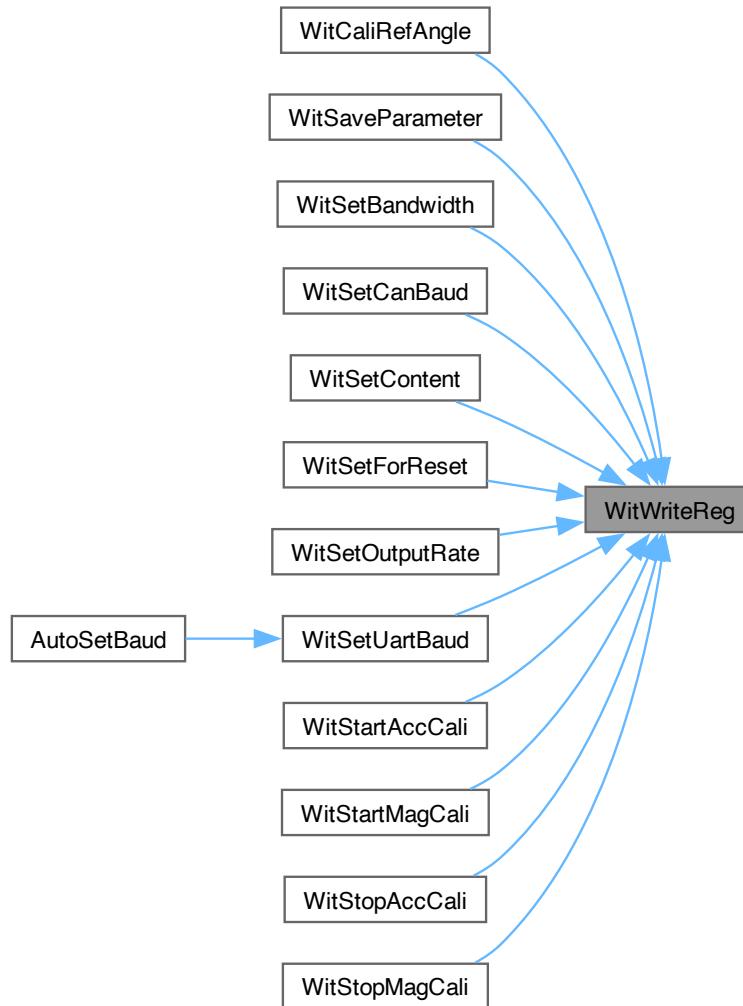
Here is the call graph for this function:



9.91.2.25 WitWriteReg()

```
int32_t WitWriteReg (
    uint32_t uiReg,
    uint16_t usData)
```

Here is the caller graph for this function:



9.91.3 Variable Documentation

9.91.3.1 __auchCRCHi

```
const uint8_t __auchCRCHi[256] [static]
```

Initial value:

```
= {
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,
    0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
    0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01,
    0xC0, 0x80, 0x41, 0x01, 0xC0, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x81,
    0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
    0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01,
    0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,
```

```

    0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
    0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01,
    0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
    0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,
    0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
    0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01,
    0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x81,
    0x40
}

```

9.91.3.2 __ auchCRCLo

```
const uint8_t __ auchCRCLo[256] [static]
```

Initial value:

```

= {
    0x00, 0xC0, 0xC1, 0x01, 0xC3, 0x03, 0x02, 0xC2, 0xC6, 0x06, 0x07, 0xC7, 0x05, 0xC5, 0xC4,
    0x04, 0xCC, 0x0C, 0x0D, 0xCD, 0x0F, 0xCF, 0xCE, 0x0E, 0x0A, 0xCA, 0xCB, 0x0B, 0xC9, 0x09,
    0x08, 0xC8, 0xD8, 0x18, 0x19, 0xD9, 0x1B, 0xDB, 0xDA, 0x1A, 0x1E, 0xDE, 0xDF, 0x1F, 0xDD,
    0x1D, 0x1C, 0xDC, 0x14, 0xD4, 0xD5, 0x15, 0xD7, 0x17, 0x16, 0xD6, 0xD2, 0x12, 0x13, 0xD3,
    0x11, 0xD1, 0xD0, 0x10, 0xF0, 0x30, 0x31, 0xF1, 0x33, 0xF3, 0xF2, 0x32, 0x36, 0xF6, 0xF7,
    0x37, 0xF5, 0x35, 0x34, 0xF4, 0x3C, 0xFC, 0xFD, 0x3D, 0xFF, 0x3F, 0x3E, 0xFE, 0xFA, 0x3A,
    0x3B, 0xFB, 0x39, 0xF9, 0xF8, 0x38, 0x28, 0xE8, 0xE9, 0x29, 0xEB, 0x2B, 0x2A, 0xEA, 0xEE,
    0x2E, 0x2F, 0xEF, 0x2D, 0xED, 0xEC, 0x2C, 0xE4, 0x24, 0x25, 0xE5, 0x27, 0xE7, 0xE6, 0x26,
    0x22, 0xE2, 0xE3, 0x23, 0xE1, 0x21, 0x20, 0xE0, 0xA0, 0x60, 0x61, 0xA1, 0x63, 0xA3, 0xA2,
    0x62, 0x66, 0xA6, 0xA7, 0x67, 0xA5, 0x65, 0x64, 0xA4, 0x6C, 0xAC, 0xAD, 0x6D, 0xAF, 0x6F,
    0x6E, 0xAE, 0xAA, 0x6A, 0x6B, 0xAB, 0x69, 0xA9, 0xA8, 0x68, 0x78, 0xB8, 0xB9, 0x79, 0xBB,
    0x7B, 0x7A, 0xBA, 0xBE, 0x7B, 0x7F, 0xBF, 0x7D, 0xBD, 0xBC, 0x7C, 0xB4, 0x74, 0x75, 0xB5,
    0x77, 0xB7, 0xB6, 0x76, 0x72, 0xB2, 0xB3, 0x73, 0xB1, 0x71, 0x70, 0xB0, 0x50, 0x90, 0x91,
    0x51, 0x93, 0x53, 0x52, 0x92, 0x96, 0x56, 0x57, 0x97, 0x55, 0x95, 0x94, 0x54, 0x9C, 0x5C,
    0x5D, 0x9D, 0x5F, 0x9F, 0x9E, 0x5E, 0x5A, 0x9A, 0x9B, 0x5B, 0x99, 0x59, 0x58, 0x98, 0x88,
    0x48, 0x49, 0x89, 0x4B, 0x8B, 0x8A, 0x4A, 0x4E, 0x8E, 0x8F, 0x4F, 0x8D, 0x4D, 0x4C, 0x8C,
    0x44, 0x84, 0x85, 0x45, 0x87, 0x47, 0x46, 0x86, 0x82, 0x42, 0x43, 0x83, 0x41, 0x81, 0x80,
    0x40
}

```

9.91.3.3 p_WitDelaymsFunc

```
DelaymsCb p_WitDelaymsFunc = NULL [static]
```

9.91.3.4 p_WitRegUpdateCbFunc

```
RegUpdateCb p_WitRegUpdateCbFunc = NULL [static]
```

9.91.3.5 p_WitSerialWriteFunc

```
SerialWrite p_WitSerialWriteFunc = NULL [static]
```

9.91.3.6 s_ucAddr

```
uint8_t s_ucAddr = 0xff [static]
```

9.91.3.7 s_ucWitDataBuff

```
uint8_t s_ucWitDataBuff[WIT_DATA_BUFF_SIZE] [static]
```

9.91.3.8 s_uiProtocolo

```
uint32_t s_uiProtocolo = 0 [static]
```

9.91.3.9 s_uiReadRegIndex

```
uint32_t s_uiReadRegIndex = 0 [static]
```

9.91.3.10 s_uiWitDataCnt

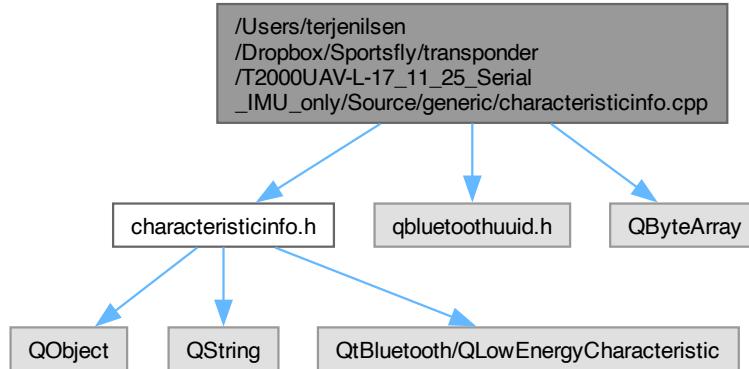
```
uint32_t s_uiWitDataCnt = 0 [static]
```

9.91.3.11 sReg

```
int16_t sReg[REGSIZE]
```

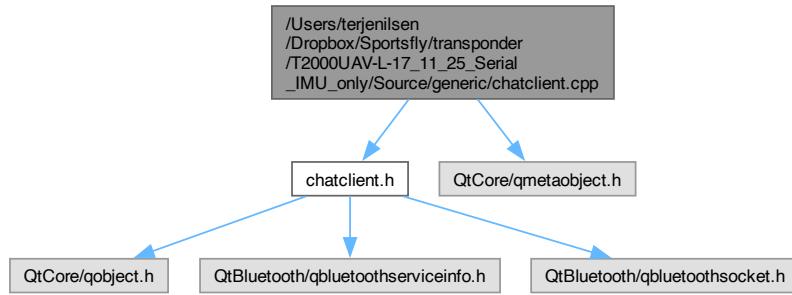
9.92 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/generic/characteristicinfo.cpp File Reference

```
#include "characteristicinfo.h"
#include "qbluetoothuuid.h"
#include <QByteArray>
Include dependency graph for characteristicinfo.cpp:
```



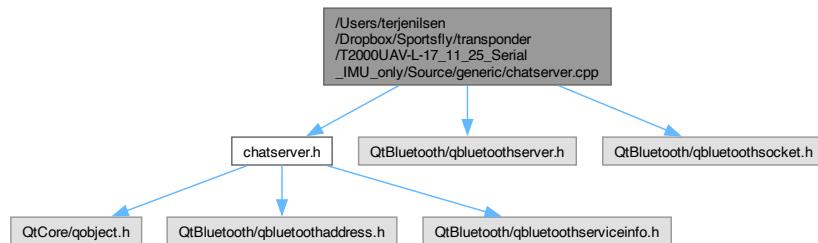
9.93 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/generic/chatclient.cpp File Reference

```
#include "chatclient.h"
#include <QtCore/qmetaobject.h>
Include dependency graph for chatclient.cpp:
```



9.94 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/generic/chatserver.cpp File Reference

```
#include "chatserver.h"
#include <QtBluetooth/qbluetoothserver.h>
#include <QtBluetooth/qbluetoothsocket.h>
Include dependency graph for chatserver.cpp:
```



Functions

- static const QLatin1String [serviceUuid \("e8e10f95-1a70-4b27-9ccf-02010264e9c8"\)](#)
[Service UUID]

9.94.1 Function Documentation

9.94.1.1 serviceUuid()

```
const QLatin1String serviceUuid ("e8e10f95-1a70-4b27-9ccf-02010264e9c8") [static]
```

[Service UUID]

Here is the caller graph for this function:



9.95 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_ ↵
11_25_Serial_IMU_only/Source/generic/device.cpp File Reference

```
#include "device.h"
#include "serviceinfo.h"
#include "characteristicinfo.h"
#include <iostream>
#include <qbluetoothaddress.h>
#include <qbluetoothdevicediscoveryagent.h>
#include <qbluetoothlocaldevice.h>
#include <qbluetoothdeviceinfo.h>
#include <qbluetoothservicediscoveryagent.h>
#include <QDebug>
#include <QList>
#include <QMetaEnum>
#include <QThread>
#include <QTimer>
#include <QtWidgets/qapplication.h>
#include <QtCore/QLoggingCategory>
#include <stdio.h>
```

Include dependency graph for device.cpp:

include dependency graph for development.



Macros

- #define key "TransponderKey"

Variables

- static QString const `channels` [] = { "{6e400002-b5a3-f393-e0a9-e50e24dcca9e}", "{6e400003-b5a3-f393-e0a9-e50e24dcca9e}" }

9.95.1 Macro Definition Documentation

9.95.1.1 key

```
#define key "TransponderKey"
```

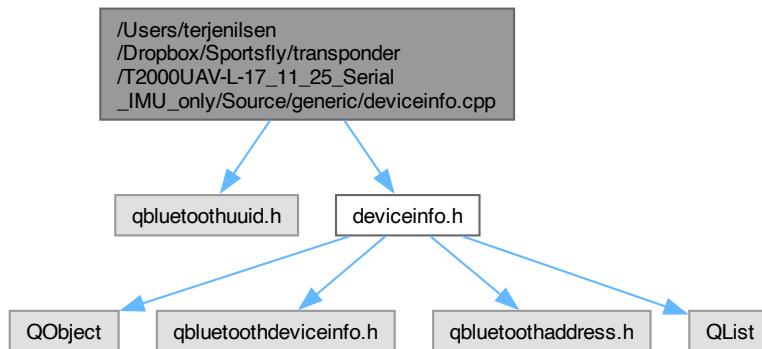
9.95.2 Variable Documentation

9.95.2.1 channels

```
QString const channels[] = { "{6e400002-b5a3-f393-e0a9-e50e24dcca9e}", "{6e400003-b5a3-f393-e0a9-e50e24dcca9e"}  
[static]
```

9.96 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/generic/deviceinfo.cpp File Reference

```
#include <qbluetoothuuid.h>
#include "deviceinfo.h"
Include dependency graph for deviceinfo.cpp:
```

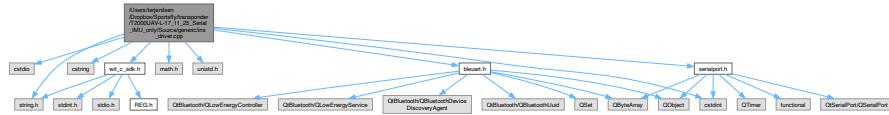


9.97 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_←
11_25_Serial_IMU_only/Source/generic/ins_driver.cpp File
Reference

Implementation of INS_driver.

```
#include <cstdio>
#include <cstdint>
#include <cstring>
#include <string.h>
#include <math.h>
#include <unistd.h>
#include "wit_c_sdk.h"
#include "bleuart.h"
#include "serialport.h"
Include dependency graph for ins_driver.cpp:
```

Include dependency graph for ins_driver.cpp:



Data Structures

- struct LFD

Typedefs

- using RxCallbackINS = void(*)(void *handler, uint32_t uiReg, uint16_t uiRegNu[])

Functions

- bool `AutoScanSensor ()`
 - void `AutoSetBaud (int)`
 - static void `SensorUartSend (uint8_t *p_data, uint32_t uiSize)`
 - static void `CopeSensorData (uint32_t uiReg, uint32_t uiRegNum)`
 - static void `setINSRxCallback (RxCallbackINS cb)`
 - bool `INS_driver (void *handler, ComQt *serPorts, ComBt *serPortb, void *func)`

Variables

- static char s_cDataUpdate = 0
 - ulong iComPort = 4
 - int iAddress = 0x50
 - static ComQt * serialPorts = nullptr
 - static ComBt * serialPortb = nullptr
 - static RxCallbackINS callback_ = nullptr
 - static void * handle = nullptr

9.97.1 Detailed Description

Implementation of INS_driver.

Contains the implementation details for the INS_driver class.

9.97.2 Typedef Documentation

9.97.2.1 RxCallbackINS

```
using RxCallbackINS = void(*)(void *handler, uint32_t uiReg, uint16_t uiRegNu[ ])
```

9.97.3 Function Documentation

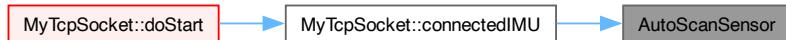
9.97.3.1 AutoScanSensor()

```
bool AutoScanSensor ()
```

Here is the call graph for this function:



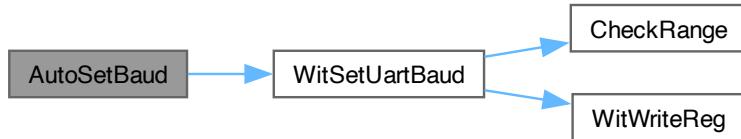
Here is the caller graph for this function:



9.97.3.2 AutoSetBaud()

```
void AutoSetBaud (
    int baud)
```

Here is the call graph for this function:



9.97.3.3 CopeSensorData()

```
void CopeSensorData (
    uint32_t uiReg,
    uint32_t uiRegNum) [static]
```

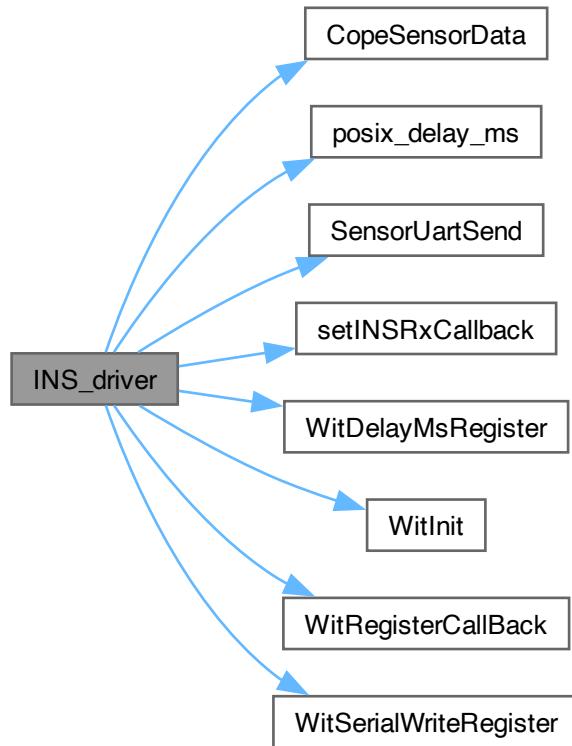
Here is the caller graph for this function:



9.97.3.4 INS_driver()

```
bool INS_driver (
    void * handler,
    ComQt * serPorts,
    ComBt * serPortb,
    void * func)
```

Here is the call graph for this function:



Here is the caller graph for this function:



9.97.3.5 SensorUartSend()

```
void SensorUartSend (
    uint8_t * p_data,
    uint32_t uiSize) [static]
```

Here is the caller graph for this function:



9.97.3.6 setINSRxCallback()

```
void setINSRxCallback (
    RxCallbackINS cb) [static]
```

Here is the caller graph for this function:



9.97.4 Variable Documentation

9.97.4.1 callback_

```
RxCallbackINS callback_ = nullptr [static]
```

9.97.4.2 handle

```
void* handle = nullptr [static]
```

9.97.4.3 iAddress

```
int iAddress = 0x50
```

9.97.4.4 iComPort

```
ulong iComPort = 4
```

9.97.4.5 s_cDataUpdate

```
char s_cDataUpdate = 0 [static]
```

9.97.4.6 serialPortb

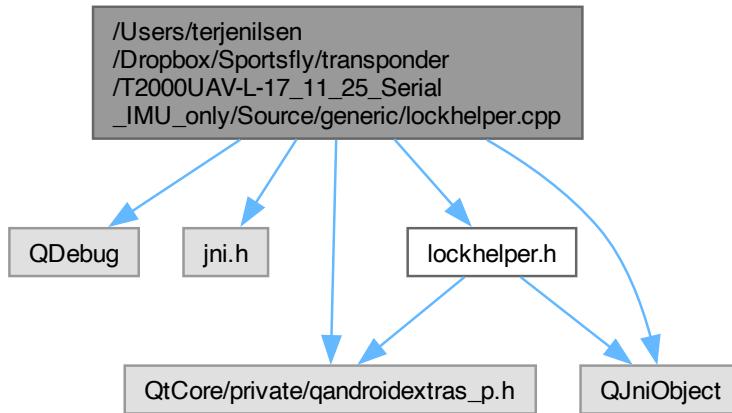
```
ComBt* serialPortb = nullptr [static]
```

9.97.4.7 serialPorts

```
ComQt* serialPorts = nullptr [static]
```

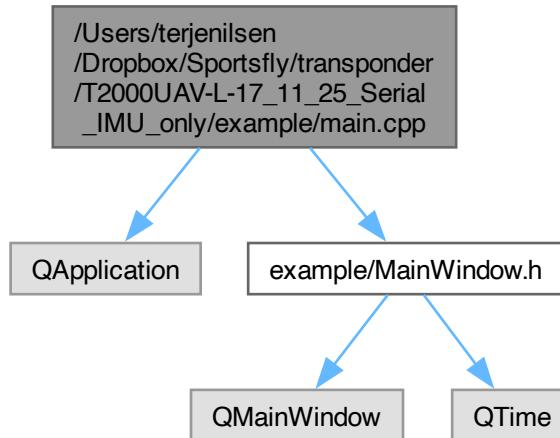
9.98 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/generic/lockhelper.cpp File Reference

```
#include <QDebug>
#include "jni.h"
#include <QtCore/private/qandroidextras_p.h>
#include <QJniObject>
#include "lockhelper.h"
Include dependency graph for lockhelper.cpp:
```



9.99 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/main.cpp File Reference

```
#include <QApplication>
#include <example/MainWindow.h>
Include dependency graph for main.cpp:
```



Functions

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

9.99.1 Function Documentation

9.99.1.1 main()

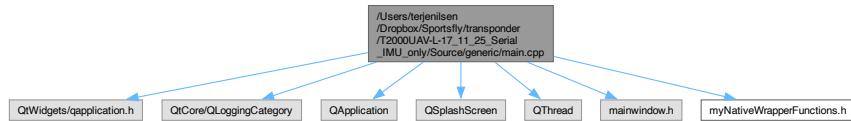
```
int main (
    int argc,
    char * argv[ ])
```

9.100 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/generic/main.cpp File Reference

Implementation of main.

```
#include <QtWidgets/qapplication.h>
#include <QtCore/QLoggingCategory>
#include <QApplication>
```

```
#include <QSplashScreen>
#include <QThread>
#include "mainwindow.h"
#include "myNativeWrapperFunctions.h"
Include dependency graph for main.cpp:
```



Functions

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

9.100.1 Detailed Description

Implementation of main.

Contains the implementation details for the main.

9.100.2 Function Documentation

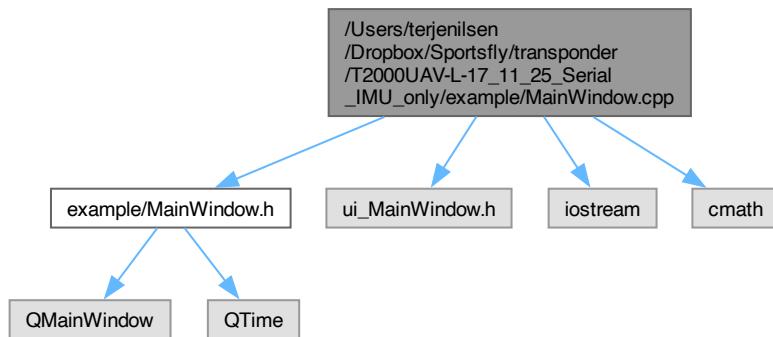
9.100.2.1 main()

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

9.101 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/example/mainwindow.cpp File Reference

```
#include <example/mainwindow.h>
#include <ui_MainWindow.h>
#include <iostream>
```

```
#include <cmath>
Include dependency graph for MainWindow.cpp:
```



9.102 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/generic/mainwindow.cpp File Reference

Implementation of [MainWindow](#).

```
#include <QtCore/QLoggingCategory>
#include <QQmlContext>
#include <QGuiApplication>
#include <QColorDialog>
#include <QNetworkInterface>
#include <QTimer>
#include <cstdio>
#include <QApplication>
#include <QEapsedTimer>
#include <QFile>
#include <QDir>
#include <QStandardPaths>
#include <QQuickWidget>
#include <QQmlProperty>
#include <QPermission>
#include <QActionGroup>
#include <QVideoWidget>
#include <QCameraDevice>
#include <QPixmap>
#include <QMediaRecorder>
#include <QImageCapture>
#include <QMediaFormat>
#include <QMediaPlayer>
#include <QOrientationSensor>
#include <QList>
#include <QSplashScreen>
#include <QtMath>
#include <deque>
```

```
#include <QThread>
#include <chrono>
#include <QtCharts/QChartView>
#include <QtCharts/QSplineSeries>
#include "mainwindow.h"
#include "gpx_parse.h"
#include "wit_c_sdk.h"
#include "geoid_helper.h"
Include dependency graph for mainwindow.cpp:
```



Macros

- #define varfilterlength 4
- #define GFILTER 100
- #define SCALE 1.00
- #define filterlength 30
- #define rotfilterlength 20
- #define headfilterlength 3

9.102.1 Detailed Description

Implementation of [MainWindow](#).

Contains the implementation details for the [MainWindow](#) class.

9.102.2 Macro Definition Documentation

9.102.2.1 filterlength

```
#define filterlength 30
```

9.102.2.2 GFILTER

```
#define GFILTER 100
```

9.102.2.3 headfilterlength

```
#define headfilterlength 3
```

9.102.2.4 rotfilterlength

```
#define rotfilterlength 20
```

9.102.2.5 SCALE

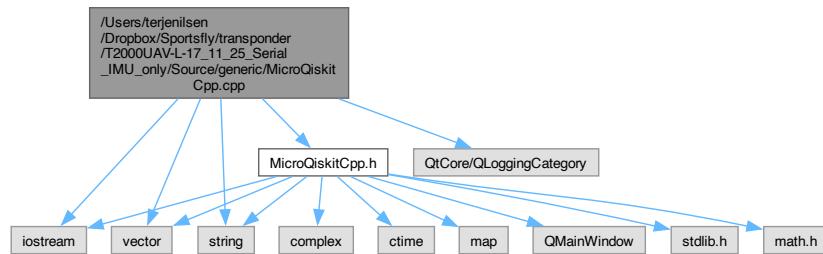
```
#define SCALE 1.00
```

9.102.2.6 varfilterlength

```
#define varfilterlength 4
```

9.103 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/generic/MicroQiskitCpp.cpp File Reference

```
#include <iostream>
#include <vector>
#include <string>
#include <QtCore/QLoggingCategory>
#include "MicroQiskitCpp.h"
Include dependency graph for MicroQiskitCpp.cpp:
```



Functions

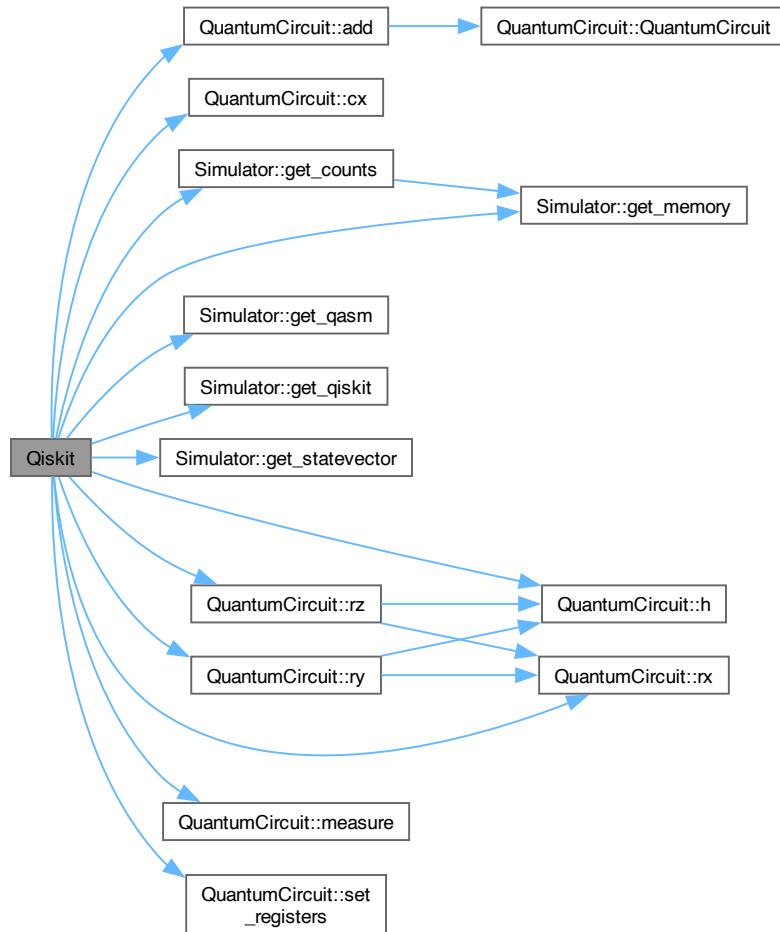
- void **Qiskit** (void)

9.103.1 Function Documentation

9.103.1.1 Qiskit()

```
void Qiskit (
    void )
```

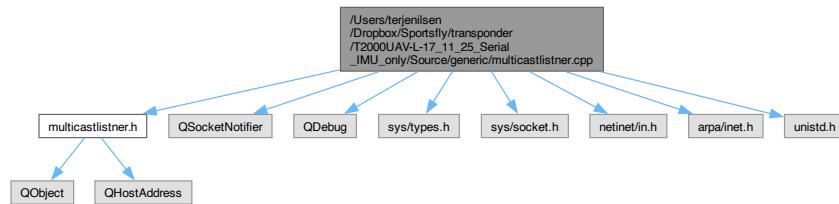
Here is the call graph for this function:



9.104 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/generic/multicastlistner.cpp File Reference

```
#include "multicastlistner.h"
#include <QSocketNotifier>
#include <QDebug>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <unistd.h>
```

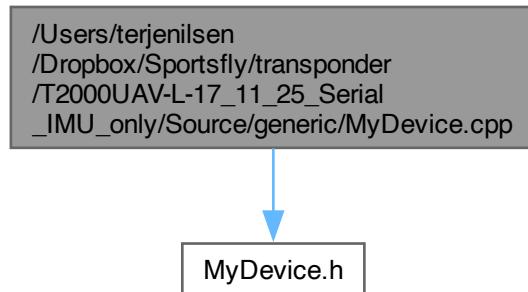
Include dependency graph for multicastlistner.cpp:



9.105 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/generic/MyDevice.cpp File Reference

```
#include "MyDevice.h"
```

Include dependency graph for MyDevice.cpp:



9.106 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/generic/mytcpsocket.cpp File Reference

Implementation of [MyTcpSocket](#).

```
#include <QTime>
#include <QTimer>
#include <QThread>
#include <QHostAddress>
#include <QNetworkInterface>
#include <QStringList>
#include <QList>
```

```
#include <QCoreApplication>
#include <QSerialPortInfo>
#include <QDebug>
#include "mytcpsocket.h"
#include "wit_c_sdk.h"
#include "tcpclient.h"
#include <QVector>
#include <QString>
#include <QMap>
#include <unistd.h>
#include "REG.h"
#include "ssdp.h"
```

Include dependency graph for mytcpsocket.cpp:



Data Structures

- struct [LFD](#)

Macros

- #define [UDP](#)

Functions

- bool [INS_driver](#) (void *, [ComQt](#) *serPorts, [ComBt](#) *serPortb, void *func)
- bool [AutoScanSensor](#) ()
- void [AutoSetBaud](#) (int)
- static int32_t [join32](#) (uint16_t lo, uint16_t hi)
Static callback from INS_driver / WIT SDK.
- static double [nmea_ddmm_to_deg](#) (int32_t raw)

9.106.1 Detailed Description

Implementation of [MyTcpSocket](#).

Contains the implementation details for the [MyTcpSocket](#) class.

9.106.2 Macro Definition Documentation

9.106.2.1 UDP

```
#define UDP
```

9.106.3 Function Documentation

9.106.3.1 AutoScanSensor()

```
bool AutoScanSensor () [extern]
```

Here is the call graph for this function:



Here is the caller graph for this function:



9.106.3.2 AutoSetBaud()

```
void AutoSetBaud (
    int baud) [extern]
```

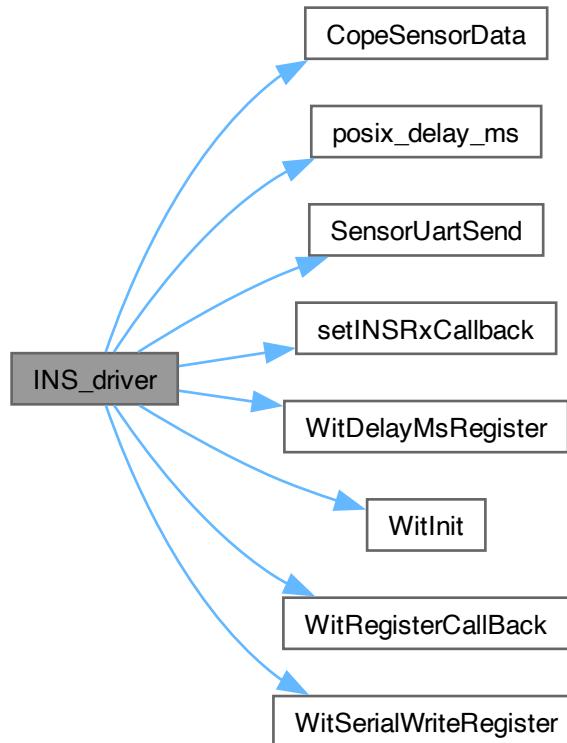
Here is the call graph for this function:



9.106.3.3 INS_driver()

```
bool INS_driver (
    void * handler,
    ComQt * serPorts,
    ComBt * serPortb,
    void * func) [extern]
```

Here is the call graph for this function:



Here is the caller graph for this function:



9.106.3.4 join32()

```
int32_t join32 (
    uint16_t lo,
    uint16_t hi) [inline], [static]
```

Static callback from INS_driver / WIT SDK.

Parses ASCII key/value pairs such as: "AccX 0.123", "AngleY 10.0", "LAT 59.00", ...

Parameters

<i>parent</i>	Pointer back to MyTcpSocket instance (this).
<i>data</i>	Null-terminated ASCII string from the IMU.
<i>length</i>	Data length in bytes (unused here).

Here is the caller graph for this function:



9.106.3.5 nmea_ddmm_to_deg()

```
double nmea_ddmm_to_deg (
    int32_t raw) [static]
```

Here is the caller graph for this function:

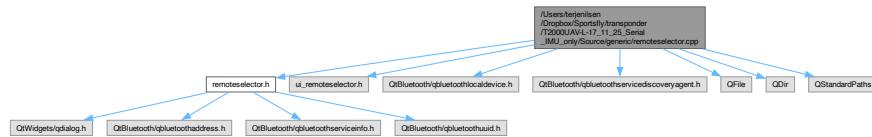


9.107 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/generic/remoteselector.cpp File Reference

```
#include "remoteselector.h"
#include "ui_remoteselector.h"
#include <QtBluetooth/qbluetoothlocaldevice.h>
#include <QtBluetooth/qbluetoothservicediscoveryagent.h>
#include <QFile>
#include <QDir>
```

```
#include <QStandardPaths>
```

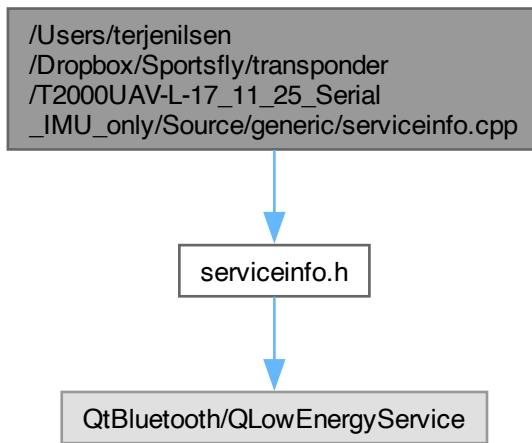
Include dependency graph for remoteselector.cpp:



9.108 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/generic/serviceinfo.cpp File Reference

```
#include "serviceinfo.h"
```

Include dependency graph for serviceinfo.cpp:

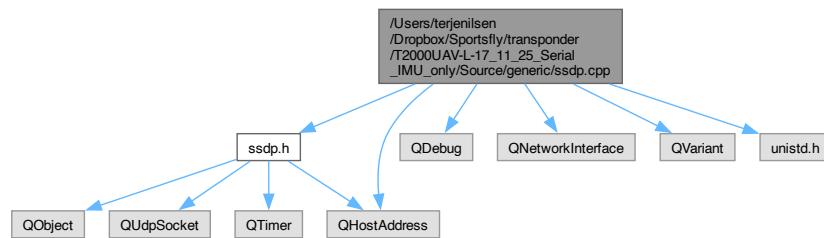


9.109 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/generic/ssdp.cpp File Reference

Implementation of SSDP discovery support.

```
#include "ssdp.h"
#include <QDebug>
#include <QHostAddress>
#include <QNetworkInterface>
#include <QVariant>
```

```
#include <unistd.h>
Include dependency graph for ssdp.cpp:
```



9.109.1 Detailed Description

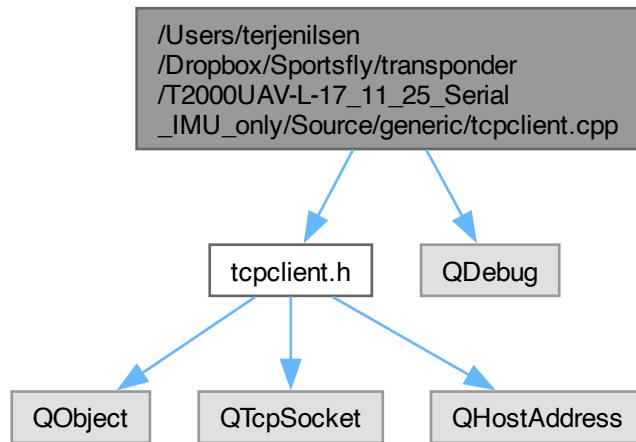
Implementation of SSDP discovery support.

Contains the internal implementation details of [SsdpDiscoverer](#), including socket setup, multicast handling, and datagram parsing.

9.110 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-L-17_11_25_Serial_IMU_only/Source/generic/tcpclient.cpp File Reference

Implementation of [TcpClient](#).

```
#include "tcpclient.h"
#include <QDebug>
Include dependency graph for tcpclient.cpp:
```



9.110.1 Detailed Description

Implementation of [TcpClient](#).

Contains the implementation details for the [TcpClient](#) class.

Index

/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-
L-17_11_25_Serial_IMU_only/Header Directory Reference, 19
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-
L-17_11_25_Serial_IMU_only/Header/driver Directory Reference, 15
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-
L-17_11_25_Serial_IMU_only/Header/driver/bleuart.h, 202, 204
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-
L-17_11_25_Serial_IMU_only/Header/driver/geoid_helper.h, 207, 208
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-
L-17_11_25_Serial_IMU_only/Header/driver/gpx_parse.h, 208, 209
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-
L-17_11_25_Serial_IMU_only/Header/driver/mqttclient.h, 209, 210
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-
L-17_11_25_Serial_IMU_only/Header/driver/rotation_matrix.h, 212, 217
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-
L-17_11_25_Serial_IMU_only/Header/driver/serialport.h, 218
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-
L-17_11_25_Serial_IMU_only/Header/driver/serialport_bleu.h, 219, 220
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-
L-17_11_25_Serial_IMU_only/Header/driver/wit_c_sdk.h, 221, 235
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-
L-17_11_25_Serial_IMU_only/Header/generic Directory Reference, 17
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-
L-17_11_25_Serial_IMU_only/Header/generic/MicroQiskit.h, 258, 260
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-
L-17_11_25_Serial_IMU_only/Header/generic/MyDevice.h, 268
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-
L-17_11_25_Serial_IMU_only/Header/generic/REG.h, 277, 306
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-
L-17_11_25_Serial_IMU_only/Header/generic/characteristic.h, 236
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-
L-17_11_25_Serial_IMU_only/Header/generic/chatclient.h, 237, 238
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-
L-17_11_25_Serial_IMU_only/Header/generic/chatserver.h, 239, 240
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-
L-17_11_25_Serial_IMU_only/Header/device.h, 241, 242
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-
L-17_11_25_Serial_IMU_only/Header/deviceinfo.h, 243, 244
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-
L-17_11_25_Serial_IMU_only/Header/lockhelper.h, 245, 246
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-
L-17_11_25_Serial_IMU_only/Header/mainwindow.h, 249, 253
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-
L-17_11_25_Serial_IMU_only/Header/multicastlistner.h, 266, 267
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-
L-17_11_25_Serial_IMU_only/Header/myNativeWrapper.h, 269, 270
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-
L-17_11_25_Serial_IMU_only/Header/mytcpsocket.h, 270, 272
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-
L-17_11_25_Serial_IMU_only/Header/remoteselector.h, 270, 271
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-
L-17_11_25_Serial_IMU_only/Header/serviceinfo.h, 312
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-
L-17_11_25_Serial_IMU_only/Header/ssdp.h, 313, 316
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-
L-17_11_25_Serial_IMU_only/Header/tcpclient.h, 316, 317
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-
L-17_11_25_Serial_IMU_only/Source Directory Reference, 20
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-
L-17_11_25_Serial_IMU_only/Source/driver Directory Reference, 16
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-
L-17_11_25_Serial_IMU_only/Source/driver/bleuart.cpp, 310
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-
L-17_11_25_Serial_IMU_only/Source/driver/characteristic.h, 310
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-
L-17_11_25_Serial_IMU_only/Source/driver/geoid_helper.cpp, 318
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-
L-17_11_25_Serial_IMU_only/Source/driver/gpco_parse.cpp,

319 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-
 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV- L-17_11_25_Serial_IMU_only/Source/generic/mytcpsocket.cpp,
 L-17_11_25_Serial_IMU_only/Source/driver/mqttclient.cpp 358
 321 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-
 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV- L-17_11_25_Serial_IMU_only/Source/generic/remoteselector.cpp
 L-17_11_25_Serial_IMU_only/Source/driver/rotation_matrix.cpp 362
 321 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-
 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV- L-17_11_25_Serial_IMU_only/Source/generic/serviceinfo.cpp,
 L-17_11_25_Serial_IMU_only/Source/driver/serialport.cpp 363
 325 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-
 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV- L-17_11_25_Serial_IMU_only/Source/generic/ssdp.cpp,
 L-17_11_25_Serial_IMU_only/Source/driver/serialport_and_bluetooth.cpp 364
 326 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-
 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV- L-17_11_25_Serial_IMU_only/Source/generic/tcpclient.cpp,
 L-17_11_25_Serial_IMU_only/Source/driver/serialport_blue_tooth.cpp,
 328 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-
 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV- L-17_11_25_Serial_IMU_only/example Directory Reference, 16
 L-17_11_25_Serial_IMU_only/Source/driver/wit_c_sdk.c, 329 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-
 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV- L-17_11_25_Serial_IMU_only/example/LayoutSquare.cpp,
 L-17_11_25_Serial_IMU_only/Source/generic 175
 Directory Reference, 18 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-
 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV- L-17_11_25_Serial_IMU_only/example/LayoutSquare.h,
 L-17_11_25_Serial_IMU_only/Source/generic/MicroQiskitOp 376
 356 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-
 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV- L-17_11_25_Serial_IMU_only/example/MainWindow.cpp,
 L-17_11_25_Serial_IMU_only/Source/generic/MyDevice.cpp 353
 358 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-
 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV- L-17_11_25_Serial_IMU_only/example/MainWindow.h,
 L-17_11_25_Serial_IMU_only/Source/generic/characteristics 247 to 256
 342 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-
 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV- L-17_11_25_Serial_IMU_only/example/WidgetAI.cpp,
 L-17_11_25_Serial_IMU_only/Source/generic/chatclient.cpp 378
 343 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-
 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV- L-17_11_25_Serial_IMU_only/example/WidgetAI.h,
 L-17_11_25_Serial_IMU_only/Source/generic/chatserver.cpp 178, 179
 343 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-
 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV- L-17_11_25_Serial_IMU_only/example/WidgetALT.cpp,
 L-17_11_25_Serial_IMU_only/Source/generic/device.cpp, 180
 344 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-
 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV- L-17_11_25_Serial_IMU_only/example/WidgetALT.h,
 L-17_11_25_Serial_IMU_only/Source/generic/deviceinfo.cpp 181, 182
 345 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-
 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV- L-17_11_25_Serial_IMU_only/example/WidgetASI.cpp,
 L-17_11_25_Serial_IMU_only/Source/generic/ins_driver.cpp 183
 346 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-
 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV- L-17_11_25_Serial_IMU_only/example/WidgetASI.h,
 L-17_11_25_Serial_IMU_only/Source/generic/lockhelper.cpp 183, 184
 351 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-
 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV- L-17_11_25_Serial_IMU_only/example/WidgetEADI.cpp,
 L-17_11_25_Serial_IMU_only/Source/generic/main.cpp, 185
 352 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-
 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV- L-17_11_25_Serial_IMU_only/example/WidgetEADI.h,
 L-17_11_25_Serial_IMU_only/Source/generic/mainwindow.cpp 186, 187
 354 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-
 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV- L-17_11_25_Serial_IMU_only/example/WidgetEHSI.cpp,
 L-17_11_25_Serial_IMU_only/Source/generic/multicastlist.cpp 189
 357 /Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV-

L-17_11_25_Serial_IMU_only/example/WidgetE~~H~~idgetTC
189, 191 MainWindow, 88
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAVWidgetVSI
L-17_11_25_Serial_IMU_only/example/WidgetHI.cppMainWindow, 88
192 ~BTComQt
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV- BTComQt, 28
L-17_11_25_Serial_IMU_only/example/WidgetHI.hChatClient
192, 194 ChatClient, 35
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAVChatServer
L-17_11_25_Serial_IMU_only/example/WidgetSix.cppChatServer, 39
194 ~ComBt
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV- ComBt, 44
L-17_11_25_Serial_IMU_only/example/WidgetSix.cppComQt
195, 196 ComQt, 50
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAVDevice
L-17_11_25_Serial_IMU_only/example/WidgetTC.cppDevice, 58
197 ~KeepAwakeHelper
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAV- KeepAwakeHelper, 67
L-17_11_25_Serial_IMU_only/example/WidgetTC.hLayoutSquare
198, 199 LayoutSquare, 69
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAVMainWindow
L-17_11_25_Serial_IMU_only/example/WidgetVSI.cppMainWindow, 79
200 ~MulticastListener
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAVMulticastListener, 96
L-17_11_25_Serial_IMU_only/example/WidgetVSI.hMyTcpSocket
200, 201 MyTcpSocket, 105
/Users/terjenilsen/Dropbox/Sportsfly/transponder/T2000UAVRemoteSelector
L-17_11_25_Serial_IMU_only/example/main.cpp, RemoteSelector, 136
352 ~WidgetAI
_IMU_id MainWindow, 87
MyTcpSocket, 116 ~WidgetALT
_CRC16 wit_c_sdk.c, 331 WidgetALT, 153
_CaliSum wit_c_sdk.c, 331 ~WidgetASI
_auchCRChi wit_c_sdk.c, 340 WidgetASI, 155
_auchCRCLo wit_c_sdk.c, 341 ~WidgetEADI
_radar_id MainWindow, 87 WidgetEADI, 157
MyTcpSocket, 116 ~WidgetEHSI
_transponder_id MainWindow, 87 WidgetEHSI, 163
MyTcpSocket, 116 ~WidgetHI
_widgetAI MainWindow, 87 WidgetHI, 166
_widgetALT MainWindow, 87 ~WidgetSix
_widgetASI MainWindow, 87 WidgetSix, 168
_widgetEADI MainWindow, 87 ~WidgetTC
_widgetEHSI MainWindow, 88 WidgetTC, 171
_widgetHI MainWindow, 88 ~WidgetVSI
_widgetTC MainWindow, 88 WidgetVSI, 173
_widgetVSI MainWindow, 88 AccelerometerRead
_widgetTC MainWindow, 88 MainWindow, 80
ACCFILT REG.h, 281
ACCRANGE REG.h, 281
ACCSENSOR REG.h, 282
AccX MyTcpSocket, 116
AccY MyTcpSocket, 116

AccZ MyTcpSocket, 116
add QuantumCircuit, 129
addcurrent MainWindow, 80
addItem LayoutSquare, 69
addnext MainWindow, 80
addWidget LayoutSquare, 69
AIRPLANE mainwindow.h, 250
ALARMLEVEL REG.h, 282
ALARMPIN REG.h, 282
ALGORITHM6 REG.h, 282
ALGORITHM9 REG.h, 282
Altimeter_data MyTcpSocket, 116
AltimeterData, 23 altitude, 23
pressure, 23
relative, 23
temperature, 23
altitude AltimeterData, 23
Altitudestat MyTcpSocket, 116
ANGLEAXIS REG.h, 282
AngleX MyTcpSocket, 116
AngleY MyTcpSocket, 117
AngleZ MyTcpSocket, 117
ANGXOFFSET REG.h, 282
ANGYOFFSET REG.h, 282
ANGZOFFSET REG.h, 282
AsX MyTcpSocket, 117
AsY MyTcpSocket, 117
AsZ MyTcpSocket, 117
AutoScanSensor ins_driver.cpp, 347
mytcpsocket.cpp, 360
AutoSetBaud ins_driver.cpp, 347
mytcpsocket.cpp, 360
AX REG.h, 282
AXIS6 REG.h, 283
AXOFFSET REG.h, 283
AY REG.h, 283
AYOFFSET REG.h, 283
AZ REG.h, 283
AZOFFSET REG.h, 283
BANDWIDTH REG.h, 283
BANDWIDTH_10HZ REG.h, 283
BANDWIDTH_184HZ REG.h, 283
BANDWIDTH_21HZ REG.h, 283
BANDWIDTH_256HZ REG.h, 284
BANDWIDTH_44HZ REG.h, 284
BANDWIDTH_5HZ REG.h, 284
BANDWIDTH_94HZ REG.h, 284
BATVAL REG.h, 284
BAUD REG.h, 284
BleUart, 24 bytesReceived, 25
connectTo, 25
dataReceived, 25
disconnected, 25
ready, 26
RxCallback, 25
scancomplete, 27
setRxCallback, 26
startScan, 26
writeBytes, 26
bleuart.h wt901Rx, 203
wt901Service, 203
wt901Tx, 203
BTComQt, 27 ~BTComQt, 28
BTComQt, 28 close, 29
dataReceived, 29
imuJavaObject, 30
open, 29
RxCallback, 28

send, 29, 30
serial_, 30
setRxCallback, 30
timerAndroid, 30
BWSCALE
 REG.h, 284
bytesReceived
 BleUart, 25
CALACCL
 REG.h, 284
CALACCR
 REG.h, 284
CALALTITUDE
 REG.h, 284
CALANGLEZ
 REG.h, 285
CALGYROACC
 REG.h, 285
CALHEXAHEDRON
 REG.h, 285
CALICNT
 REG.h, 285
callback
 MyTcpSocket::Callbacks, 31
callback_
 ComBt, 47
 ComQt, 54
 ins_driver.cpp, 350
Callbacks
 MyTcpSocket, 103
callbacks
 MyTcpSocket, 117
CallSwiftMemberFromC
 MyTcpSocket, 105
CALMAG
 REG.h, 285
CALMAG2STEP
 REG.h, 285
CALMAGMM
 REG.h, 285
CALREFANGLE
 REG.h, 285
CALSW
 REG.h, 285
CAN_BAUD_10000
 REG.h, 285
CAN_BAUD_100000
 REG.h, 286
CAN_BAUD_1000000
 REG.h, 286
CAN_BAUD_125000
 REG.h, 286
CAN_BAUD_20000
 REG.h, 286
CAN_BAUD_200000
 REG.h, 286
CAN_BAUD_250000
 REG.h, 286
CAN_BAUD_3000
 REG.h, 286
CAN_BAUD_40000
 REG.h, 286
CAN_BAUD_400000
 REG.h, 286
CAN_BAUD_5000
 REG.h, 286
CAN_BAUD_50000
 REG.h, 287
CAN_BAUD_500000
 REG.h, 287
CAN_BAUD_80000
 REG.h, 287
CAN_BAUD_800000
 REG.h, 287
CCBW
 REG.h, 287
ch
 QuantumCircuit, 129
channels
 device.cpp, 345
characteristicChanged
 CharacteristicInfo, 33
CharacteristicInfo, 32
 characteristicChanged, 33
 CharacteristicInfo, 32
 getCharacteristic, 33
 getName, 33
 getPermission, 33
 getUuid, 33
 getValue, 33
 setCharacteristic, 34
characteristicsUpdated
 Device, 58
ChatClient, 34
 ~ChatClient, 35
 ChatClient, 35
 connected, 36
 disconnected, 36
 messageReceived, 36
 sendMessage, 37
 socketErrorOccurred, 37
 startClient, 37
 stopClient, 37
ChatServer, 38
 ~ChatServer, 39
 ChatServer, 39
 clientConnected, 40
 clientDisconnected, 40
 messageReceived, 40
 sendMessage, 40
 startServer, 41
 stopServer, 41
chatserver.cpp
 serviceUuid, 344
CheckRange
 wit_c_sdk.c, 331

wit_c_sdk.h, 226
CHIPIDH
 REG.h, 287
CHIPIDL
 REG.h, 287
classPtr
 MyTcpSocket::Callbacks, 31
client
 MqttClient, 94
CLIENT_ID
 MyTcpSocket, 117
clientConnected
 ChatServer, 40
clientDisconnected
 ChatServer, 40
close
 BTComQt, 29
 ComBt, 45
 ComQt, 50
com_setup
 MyTcpSocket, 105
ComBt, 42
 ~ComBt, 44
 callback_, 47
 close, 45
 ComBt, 44
 connectionChanged, 45
 dataReceived, 45
 errorReceived, 45
 open, 45
 parent, 47
 RxCallback, 44
 scanDone, 47
 send, 46
 serial_, 47
 setBaudrate, 46
 setRxCallback, 46
 status, 47
compensatedHeight
 GeoidHelper, 65
ComQt, 48
 ~ComQt, 50
 callback_, 54
 close, 50
 ComQt, 50
 connectionChanged, 50
 dataReceived, 51
 errorReceived, 51
 isOpen, 52
 open, 52
 parent, 54
 RxCallback, 49
 send, 52, 53
 serial_, 54
 setBaudrate, 53
 setRxCallback, 54
 status, 54
CONFIG
 mainwindow.h, 250
connect
 MqttClient, 94
connected
 ChatClient, 36
 MyTcpSocket, 105
 TcpClient, 147
connectedAltitude
 MyTcpSocket, 106
connectedIMU
 MyTcpSocket, 107
connectedRadar
 MyTcpSocket, 108
connectionChanged
 ComBt, 45
 ComQt, 50
connectTo
 BleUart, 25
 TcpClient, 147
connectToService
 Device, 58
ConnListener, 55
 ConnListener, 56
 on_failure, 56
 on_success, 56
CopeSensorData
 ins_driver.cpp, 348
CopeWitData
 wit_c_sdk.c, 331
count
 LayoutSquare, 70
createRotationMatrix
 rotation_matrix.cpp, 322
 rotation_matrix.h, 215
createRotationMatrixZ
 rotation_matrix.cpp, 323
 rotation_matrix.h, 215
crx
 QuantumCircuit, 130
current
 MainWindow, 88
currentIndex
 MainWindow, 88
cx
 QuantumCircuit, 130
D
 LFD, 74
D0MODE
 REG.h, 287
D0PWMH
 REG.h, 287
D0PWMT
 REG.h, 287
D0Status
 REG.h, 288
D1MODE
 REG.h, 288
D1PWMH

REG.h, 288
D1PWMT
 REG.h, 288
D1Status
 REG.h, 288
D2MODE
 REG.h, 288
D2PWMH
 REG.h, 288
D2PWMT
 REG.h, 288
D2Status
 REG.h, 288
D3MODE
 REG.h, 288
D3PWMH
 REG.h, 289
D3PWMT
 REG.h, 289
D3Status
 REG.h, 289
data
 QuantumCircuit, 134
dataReceived
 BleUart, 25
 BTComQt, 29
 ComBt, 45
 ComQt, 51
 TcpClient, 147
DDHH
 REG.h, 289
DEG_TO_RAD
 rotation_matrix.h, 213
DelaymsCb
 wit_c_sdk.h, 226
DELAYT
 REG.h, 289
description
 PortEntry, 126
Device, 57
 ~Device, 58
 characteristicsUpdated, 58
 connectToService, 58
 Device, 58
 disconnected, 59
 disconnectFromDevice, 59
 key2, 62
 l_service, 62
 m_message, 62
 randomAddressChanged, 59
 receivedData, 62
 scanServices, 59
 SendData, 59
 setVal, 60
 startDeviceDiscovery, 60
 state, 60
 stateChanged, 60
 test, 61
updateChanged, 61
updateDisplay, 61
updateValueSet, 61
device.cpp
 channels, 345
 key, 345
deviceChanged
 DeviceInfo, 63
deviceFound
 SsdpDiscoverer, 145
DeviceInfo, 62
 deviceChanged, 63
 DeviceInfo, 63
 deviceName, 64
 getAddress, 63
 getDevice, 64
 getName, 64
 setDevice, 64
deviceName
 DeviceInfo, 64
disc
 MyTcpSocket, 117
disconnect
 MqttClient, 94
disconnected
 BleUart, 25
 ChatClient, 36
 Device, 59
 TcpClient, 147
disconnectFrom
 TcpClient, 147
disconnectFromDevice
 Device, 59
doConnect
 MyTcpSocket, 109
Donwn_Speed
 MyTcpSocket, 118
doRadar
 MyTcpSocket, 109
doStart
 MyTcpSocket, 110
doTransponder
 MyTcpSocket, 111
dt
 TrackPoint, 148
EARTH_RADIUS
 gpx_parse.cpp, 321
elevation
 TrackPoint, 148
EnableKeepAwakeHelper
 KeepAwakeHelper, 67
ERROR
 MicroQiskitCpp.h, 260
error_handler
 MicroQiskitCpp.h, 260
errorOccurred
 MulticastListener, 97
 TcpClient, 147

errorReceived
 ComBt, 45
 ComQt, 51
ETX
 mytcpsocket.h, 272
expandingDirections
 LayoutSquare, 70

F
 LFD, 74
filterlength
 mainwindow.cpp, 355
FILTK
 REG.h, 289
FLIGHTLOG
 mainwindow.h, 250
FromID
 MyTcpSocket, 118
FuncR
 wit_c_sdk.c, 330
FuncW
 wit_c_sdk.c, 330

G
 MyTcpSocket, 118
GeoidHelper, 64
 compensatedHeight, 65
 GeoidHelper, 65
 isValid, 65
GeoidHelper::Impl, 67
 valid, 67
GeoidResult, 65
 h_compensated, 65
 N, 65
geometry2
 LayoutSquare, 70
get_counts
 Simulator, 141
get_default_config
 MainWindow, 80
get_memory
 Simulator, 141
get_qasm
 Simulator, 142
get_qiskit
 Simulator, 142
get_statevector
 Simulator, 142
getActiveScreen
 MainWindow, 81
getAddress
 DeviceInfo, 63
getAI
 MainWindow, 81
 WidgetSix, 168
getALT
 MainWindow, 81
 WidgetSix, 168
getASI

 MainWindow, 81
 WidgetSix, 168
getBearing
 MainWindow, 81
getCharacteristic
 CharacteristicInfo, 33
getDevice
 DeviceInfo, 64
getEADI
 MainWindow, 81
 WidgetSix, 169
getEHJSI
 MainWindow, 82
getHI
 MainWindow, 82
 WidgetSix, 169
getName
 CharacteristicInfo, 33
 DeviceInfo, 64
 ServiceInfo, 139
getPermission
 CharacteristicInfo, 33
getSelf
 serialport_android.cpp, 326
getTC
 MainWindow, 82
 WidgetSix, 169
getTrackPoints
 GpxParser, 66
getType
 ServiceInfo, 139
getUuid
 CharacteristicInfo, 33
 ServiceInfo, 139
getValue
 CharacteristicInfo, 33
getVSI
 MainWindow, 82
 WidgetSix, 169
GFILTER
 mainwindow.cpp, 355
GPSBAUD
 REG.h, 289
GPSHeight
 REG.h, 289
GPSTYPE
 REG.h, 289
GPSVH
 REG.h, 289
GPSVL
 REG.h, 290
GPSVX_H
 REG.h, 290
GPSVX_L
 REG.h, 290
GPSVY_H
 REG.h, 290
GPSVY_L

REG.h, 290
GPSVZ_H
 REG.h, 290
GPSVZ_L
 REG.h, 290
GPSYAW
 REG.h, 290
gpx_parse.cpp
 EARTH_RADIUS, 321
 haversineDistance, 320
GpxParser, 66
 getTrackPoints, 66
 parseFile, 66
GX
 REG.h, 290
GXOFFSET
 REG.h, 290
GY
 REG.h, 291
GYOFFSET
 REG.h, 291
GYROCALTHR
 REG.h, 291
GYROCALTIME
 REG.h, 291
GYROFILT
 REG.h, 291
GYRORANGE
 REG.h, 291
GYROSENSOR
 REG.h, 291
GYROZSCALE
 REG.h, 291
GZ
 REG.h, 291
GZOFFSET
 REG.h, 291
h
 QuantumCircuit, 130
h_compensated
 GeoidResult, 65
handle
 ins_driver.cpp, 350
handleUpdate
 MyTcpSocket, 111
has_measurements
 QuantumCircuit, 130
hasHeightForWidth
 LayoutSquare, 70
hasItem
 LayoutSquare, 70
haversineDistance
 gpx_parse.cpp, 320
HDOP
 REG.h, 292
headfilterlength
 mainwindow.cpp, 355
HeightH
 REG.h, 292
 HeightL
 REG.h, 292
 hideCamera
 MainWindow, 82
 HPitch
 REG.h, 292
 HRoll
 REG.h, 292
 HX
 MyTcpSocket, 118
 HXi
 REG.h, 292
 HXOFFSET
 REG.h, 292
 HY
 MyTcpSocket, 118
 HYaw
 REG.h, 292
 HYi
 REG.h, 292
 HYOFFSET
 REG.h, 292
 HZ
 MyTcpSocket, 118
 HZi
 REG.h, 293
 HZOFFSET
 REG.h, 293
iAddress
 ins_driver.cpp, 350
iComPort
 ins_driver.cpp, 350
IICADDR
 REG.h, 293
IMAGES_DIR
 mainwindow.h, 250
IMUconnected
 MyTcpSocket, 118
imuData
 MyTcpSocket, 118
imuJavaObject
 BTComQt, 30
init
 MainWindow, 82
initialize
 QuantumCircuit, 131
INS_driver
 ins_driver.cpp, 348
 mytcpsocket.cpp, 360
ins_driver.cpp
 AutoScanSensor, 347
 AutoSetBaud, 347
 callback_, 350
 CopeSensorData, 348
 handle, 350
 iAddress, 350
 iComPort, 350

INS_driver, 348
 RxCallbackINS, 347
 s_cDataUpdate, 350
 SensorUartSend, 349
 serialPortb, 351
 serialPorts, 351
 setINSRxCallback, 350
 isOpen
 ComQt, 52
 isValid
 GeoidHelper, 65
 itemAt
 LayoutSquare, 71
 Java_com_hoho_android_usbserial_driver_TestClassTerjeLOG_ON
 serialport_android.cpp, 327
 Java_com_hoho_android_usbserial_driver_TestClassTerjeLOG_OnSerialBytes
 serialport_android.cpp, 327
 Java_com_hoho_android_usbserial_driver_TestClassTerjeLOG_OnSerialError
 serialport_android.cpp, 328
 join32
 mytcpsocket.cpp, 361
 KeepAwakeHelper, 67
 ~KeepAwakeHelper, 67
 EnableKeepAwakeHelper, 67
 KeepAwakeHelper, 67
 KEY
 REG.h, 293
 key
 device.cpp, 345
 key2
 Device, 62
 KEY_UNLOCK
 REG.h, 293
 L
 LFD, 74
 l_service
 Device, 62
 LatH
 REG.h, 293
 latitude
 TrackPoint, 148
 LatL
 REG.h, 293
 LayoutSquare, 68
 ~LayoutSquare, 69
 addItem, 69
 addWidget, 69
 count, 70
 expandingDirections, 70
 geometry2, 70
 hasHeightForWidth, 70
 hasItem, 70
 itemAt, 71
 LayoutSquare, 69
 minimumSize, 71
 replaceItem, 71
 setGeometry, 72
 sizeHint, 72
 take, 73
 takeAt, 73
 LayoutSquare.cpp
 margin, 175
 LEDOFF
 REG.h, 293
 LFD, 74
 D, 74
 F, 74
 L, 74
 lidar
 MyTcpSocket, 119
 LOG_ON
 mainwindow.h, 250
 logLanded
 MainWindow, 82
 logLeave
 MainWindow, 83
 longitude
 TrackPoint, 148
 LonH
 REG.h, 293
 LonL
 REG.h, 293
 LPitch
 REG.h, 294
 LRoll
 REG.h, 294
 LYaw
 REG.h, 294
 m_address
 MyTcpSocket, 119
 m_altitude
 MyTcpSocket, 119
 m_attitude
 MainWindow, 88
 m_bluetoothrunning
 MainWindow, 88
 m_external
 MyTcpSocket, 119
 m_gpsbearing
 MyTcpSocket, 119
 m_gpsspeed
 MyTcpSocket, 119
 m_has_MQTT
 MyTcpSocket, 119
 m_has_MQTT_accel
 MyTcpSocket, 119
 m_has_MQTT_airspeed
 MyTcpSocket, 120
 m_has_MQTT_gyro
 MyTcpSocket, 120
 m_has_MQTT_heading
 MyTcpSocket, 120
 m_has_MQTT_pressure
 MyTcpSocket, 120

m_has_MQTT_vsi
 MyTcpSocket, 120
m_head
 MainWindow, 88
m_imu_setup_done
 MyTcpSocket, 120
m_install
 MainWindow, 89
m_landedTime
 MainWindow, 89
m_latitude
 MyTcpSocket, 120
m_longitude
 MyTcpSocket, 120
m_message
 Device, 62
m_ms
 MainWindow, 89
m_msgBox
 MainWindow, 89
M_PI
 rotation_matrix.h, 213
m_preshape
 MyTcpSocket, 120
m_preshape_alt
 MyTcpSocket, 121
m_preshape_QNH
 MyTcpSocket, 121
m_pressure_raw
 MyTcpSocket, 121
m_roll_angle
 MainWindow, 89
m_speed
 MyTcpSocket, 121
m_takeoffTime
 MainWindow, 89
m_tansALT
 MainWindow, 89
m_temp
 MainWindow, 89
m_timer
 MainWindow, 90
m_total_accel
 MainWindow, 90
m_use_imu
 MainWindow, 90
m_var_speed
 MainWindow, 90
m_vario
 MainWindow, 90
m_vel_active
 MyTcpSocket, 121
m_vel_D
 MyTcpSocket, 121
m_vel_E
 MyTcpSocket, 121
m_vel_N
 MyTcpSocket, 121

MAGFILT
 REG.h, 294
MAGRANGX
 REG.h, 294
MAGRANGY
 REG.h, 294
MAGRANGZ
 REG.h, 294
MAGSENSOR
 REG.h, 294
main
 main.cpp, 352, 353
main.cpp
 main, 352, 353
MainWindow
 74
 _IMU_id, 87
 _radar_id, 87
 _transponder_id, 87
 _widgetAI, 87
 _widgetALT, 87
 _widgetASI, 87
 _widgetEADI, 87
 _widgetEHSI, 88
 _widgetHI, 88
 _widgetTC, 88
 _widgetVSI, 88
 ~MainWindow, 79
AccelerometerRead, 80
addcurrent, 80
addnext, 80
current, 88
currentIndex, 88
get_default_config, 80
getActiveScreen, 81
getAI, 81
getALT, 81
getASI, 81
getBearing, 81
getEADI, 81
getEHJSI, 82
getHI, 82
getTC, 82
getVSI, 82
hideCamera, 82
init, 82
logLanded, 82
logTakeoff, 83
m_attitude, 88
m_bluetoothrunning, 88
m_head, 88
m_install, 89
m_landedTime, 89
m_ms, 89
m_msgBox, 89
m_roll_angle, 89
m_takeoffTime, 89
m_tansALT, 89
m_temp, 89

m_timer, 90
 m_total_accel, 90
 m_use_imu, 90
 m_var_speed, 90
 m_vario, 90
 MainWindow, 78, 79
 mode, 90
 mysocket, 90
 next, 90
 onResized, 83
 permissionUpdated, 83
 Radar_depth, 91
 Radar_Height, 91
 roll_blended, 91
 roll_blended_ok, 91
 rotationMatrix, 91
 screen_index, 91
 ScreenMode, 91
 series, 91
 set_default_config, 83
 set_default_planes, 84
 set_default_radio, 84
 setButtonIcon, 84
 setCamera, 85
 setIMU, 85
 setmode, 85
 setQNH, 86
 showImage, 86
 splash, 92
 takeoff_altitude, 92
 takeoff_latitude, 92
 takeoff_longitude, 92
 timerEvent, 86
 ui, 92
 updateCameras, 86
 view, 92
 mainwindow.cpp
 filterlength, 355
 GFILTER, 355
 headfilterlength, 355
 rotfilterlength, 355
 SCALE, 355
 varfilterlength, 356
 mainwindow.h
 AIRPLANE, 250
 CONFIG, 250
 FLIGHTLOG, 250
 IMAGES_DIR, 250
 LOG_DIR, 250
 P_AUTOPILOT, 251
 P_CAMERA, 251
 P_CONFIG, 251
 P_FLIGT_INSTRUMENT, 251
 P_GLASS_COCPIT, 251
 P_IMU, 251
 P_RADAR, 251
 P_RADIO_LIST, 251
 P_TRANSPOUNDER, 251
 Qiskit, 252
 RADIO, 251
 TRANSPONDERLOG, 252
 USE_KeepAwakeHelper, 252
 manufacturer
 PortEntry, 126
 margin
 LayoutSquare.cpp, 175
 Matrix3x3
 rotation_matrix.h, 214
 Matrix3x6
 rotation_matrix.h, 214
 Matrix9x9
 rotation_matrix.h, 214
 MCAST_PORT
 MyTcpSocket, 122
 measure
 QuantumCircuit, 131
 messageReceived
 ChatClient, 36
 ChatServer, 40
 MulticastListener, 97
 MicroQiskitCpp.cpp
 Qiskit, 356
 MicroQiskitCpp.h
 ERROR, 260
 error_handler, 260
 Qt, 260
 RED, 260
 RESET, 260
 minimumSize
 LayoutSquare, 71
 MMSS
 REG.h, 294
 MODDELAY
 REG.h, 294
 mode
 MainWindow, 90
 MOVESTA
 REG.h, 295
 MOVETHR
 REG.h, 295
 mqtt
 MyTcpSocket, 122
 MqttClient, 92
 client, 94
 connect, 94
 disconnect, 94
 MqttClient, 94
 MsgHandler, 93
 sendMessage, 94
 setMessageHandler, 95
 subscribe, 95
 MS
 REG.h, 295
 MsgHandler
 MqttClient, 93
 MulticastListener, 96

~MulticastListener, 96
errorOccurred, 97
messageReceived, 97
MulticastListener, 96
start, 97
stop, 98
multiplyMatrix
 rotation_matrix.cpp, 323
 rotation_matrix.h, 215
MyDevice, 98
 MyDevice, 99
myNativeWrapperFunctions.h
 setlosParams, 269
mysocket
 MainWindow, 90
MyTcpSocket, 99
 _IMU_id, 116
 _radar_id, 116
 _transponder_id, 116
 ~MyTcpSocket, 105
 AccX, 116
 AccY, 116
 AccZ, 116
 Altimeter_data, 116
 Altitudestat, 116
 AngleX, 116
 AngleY, 117
 AngleZ, 117
 AsX, 117
 AsY, 117
 AsZ, 117
 Callbacks, 103
 callbacks, 117
 CallSwiftMemberFromC, 105
 CLIENT_ID, 117
 com_setup, 105
 connected, 105
 connectedAltitude, 106
 connectedIMU, 107
 connectedRadar, 108
 disc, 117
 doConnect, 109
 Donwn_Speed, 118
 doRadar, 109
 doStart, 110
 doTransponder, 111
 FromID, 118
 G, 118
 handleUpdate, 111
 HX, 118
 HY, 118
 HZ, 118
 IMUconnected, 118
 imuData, 118
 lidar, 119
 m_address, 119
 m_altitude, 119
 m_external, 119
 m_gpsbearing, 119
 m_gpsspeed, 119
 m_has_MQTT, 119
 m_has_MQTT_accel, 119
 m_has_MQTT_airspeed, 120
 m_has_MQTT_gyro, 120
 m_has_MQTT_heading, 120
 m_has_MQTT_pressure, 120
 m_has_MQTT_vsi, 120
 m_imu_setup_done, 120
 m_latitude, 120
 m_longitude, 120
 m_pressure, 120
 m_pressure_alt, 121
 m_pressure_QNH, 121
 m_pressure_raw, 121
 m_speed, 121
 m_vel_active, 121
 m_vel_D, 121
 m_vel_E, 121
 m_vel_N, 121
 MCAST_PORT, 122
 mqtt, 122
 MyTcpSocket, 104
 Orient, 122
 parseAltimeterLine, 112
 parseIMU, 112
 port, 122
 Radarstat, 122
 rDist, 122
 readyWrite, 113
 ret_imu, 122
 ret_transponder, 122
 rPos, 123
 rSpeed, 123
 sendMessage, 114
 serialport, 123
 SERVER_ADDRESS, 123
 setbacklit, 114
 setSerialPorts, 114
 sport, 123
 ssdpConfig, 115
 Temp, 123
 text, 123
 Transponderstat, 123
 TransponderstatWithBarometer, 124
 use_ins_only, 124
 VER, 124
mytcpsocket.cpp
 AutoScanSensor, 360
 AutoSetBaud, 360
 INS_driver, 360
 join32, 361
 nmea_ddmm_to_deg, 362
 UDP, 359
mytcpsocket.h
 ETX, 272
 SCREEN, 271

simGPS, 271
 STX, 272
 USE_BT_IMU, 271
 MyTcpSocket::Callbacks, 31
 callback, 31
 classPtr, 31

N

GeoidResult, 65
 name
 SensorService, 137
 nBits
 QuantumCircuit, 134
 next
 MainWindow, 90
 nmea_ddmm_to_deg
 mytcpsocket.cpp, 362
 NoButtonMessageBox, 124
 NoButtonMessageBox, 125
 setText, 125
 NORMAL
 REG.h, 295
 nQubits
 QuantumCircuit, 135
 NUMBERID1
 REG.h, 295
 NUMBERID2
 REG.h, 295
 NUMBERID3
 REG.h, 295
 NUMBERID4
 REG.h, 295
 NUMBERID5
 REG.h, 295
 NUMBERID6
 REG.h, 295

on_failure

ConnListener, 56

on_success

ConnListener, 56

onResized

MainWindow, 83

open

BTComQt, 29
 ComBt, 45
 ComQt, 52

ORIENT

REG.h, 296

Orient

MyTcpSocket, 122

ORIENT_HORIZONE

REG.h, 296

ORIENT_VERTICLE

REG.h, 296

P_AUTOPILOT

mainwindow.h, 251

P_CAMERA

mainwindow.h, 251

P_CONFIG

mainwindow.h, 251

P_FLIGT_INSTRUMENT

mainwindow.h, 251

P_GLASS_COCPIT

mainwindow.h, 251

P_IMU

mainwindow.h, 251

P_RADAR

mainwindow.h, 251

P_RADIO_LIST

mainwindow.h, 251

P_TRANSPODER

mainwindow.h, 251

p_WitDelaymsFunc

wit_c_sdk.c, 341

p_WitRegUpdateCbFunc

wit_c_sdk.c, 341

p_WitSerialWriteFunc

wit_c_sdk.c, 341

parent

ComBt, 47
 ComQt, 54

parseAltimeterLine

MyTcpSocket, 112

parseFile

GpxParser, 66

parseIMU

MyTcpSocket, 112

PDOP

REG.h, 296

permissionUpdated

MainWindow, 83

Pitch

REG.h, 296

port

MyTcpSocket, 122

PortEntry, 126

description, 126
 manufacturer, 126
 portName, 127
 productId, 127
 serial, 127
 systemLocation, 127
 vendorId, 127

portName

PortEntry, 127

posix_delay_ms

wit_c_sdk.c, 332
 wit_c_sdk.h, 226

POWONSEND

REG.h, 296

PRESSENSOR

REG.h, 296

pressure

AltimeterData, 23

PressureH

REG.h, 296
PressureL
 REG.h, 296
printVector
 rotation_matrix.cpp, 323
 rotation_matrix.h, 215
productId
 PortEntry, 127

q0
 REG.h, 296
q1
 REG.h, 297
q2
 REG.h, 297
q3
 REG.h, 297
qc
 Simulator, 143
Qiskit
 mainwindow.h, 252
 MicroQiskitCpp.cpp, 356
Qt
 MicroQiskitCpp.h, 260
QuantumCircuit, 127
 add, 129
 ch, 129
 crx, 130
 cx, 130
 data, 134
 h, 130
 has_measurements, 130
 initialize, 131
 measure, 131
 nBits, 134
 nQubits, 135
 QuantumCircuit, 128
 rx, 131
 ry, 131
 rz, 132
 set_registers, 133
 x, 133
 y, 133
 z, 134

RAD_TO_DEG
 rotation_matrix.h, 214
Radar_depth
 MainWindow, 91
Radar_Height
 MainWindow, 91
Radarstat
 MyTcpSocket, 122
RADIO
 mainwindow.h, 251
randomAddressChanged
 Device, 59
rDist
 MyTcpSocket, 122

READADDR
 REG.h, 297
ready
 BleUart, 26
readyWrite
 MyTcpSocket, 113
receivedData
 Device, 62
RED
 MicroQiskitCpp.h, 260
redraw
 WidgetAI, 151
 WidgetALT, 153
 WidgetASI, 155
 WidgetEADI, 158
 WidgetEHSI, 163
 WidgetHI, 166
 WidgetTC, 171
 WidgetVSI, 173
REFPITCH
 REG.h, 297
REFROLL
 REG.h, 297
REFYAW
 REG.h, 297
REG.h
 ACCFILT, 281
 ACCRANGE, 281
 ACCSENSOR, 282
 ALARMLEVEL, 282
 ALARMPIN, 282
 ALGRITHM6, 282
 ALGRITHM9, 282
 ANGLEAXIS, 282
 ANGXOFFSET, 282
 ANGYOFFSET, 282
 ANGZOFFSET, 282
 AX, 282
 AXIS6, 283
 AXOFFSET, 283
 AY, 283
 AYOFFSET, 283
 AZ, 283
 AZOFFSET, 283
 BANDWIDTH, 283
 BANDWIDTH_10HZ, 283
 BANDWIDTH_184HZ, 283
 BANDWIDTH_21HZ, 283
 BANDWIDTH_256HZ, 284
 BANDWIDTH_44HZ, 284
 BANDWIDTH_5HZ, 284
 BANDWIDTH_94HZ, 284
 BATVAL, 284
 BAUD, 284
 BWSCALE, 284
 CALACCL, 284
 CALACCR, 284
 CALALTITUDE, 284

CALANGLEZ, 285
CALGYROACC, 285
CALHEXAHEDRON, 285
CALICNT, 285
CALMAG, 285
CALMAG2STEP, 285
CALMAGMM, 285
CALREFANGLE, 285
CALSW, 285
CAN_BAUD_10000, 285
CAN_BAUD_100000, 286
CAN_BAUD_1000000, 286
CAN_BAUD_125000, 286
CAN_BAUD_20000, 286
CAN_BAUD_200000, 286
CAN_BAUD_250000, 286
CAN_BAUD_3000, 286
CAN_BAUD_40000, 286
CAN_BAUD_400000, 286
CAN_BAUD_5000, 286
CAN_BAUD_50000, 287
CAN_BAUD_500000, 287
CAN_BAUD_80000, 287
CAN_BAUD_800000, 287
CCBW, 287
CHIPIDH, 287
CHIPIDL, 287
D0MODE, 287
D0PWMH, 287
D0PWMT, 287
D0Status, 288
D1MODE, 288
D1PWMH, 288
D1PWMT, 288
D1Status, 288
D2MODE, 288
D2PWMH, 288
D2PWMT, 288
D2Status, 288
D3MODE, 288
D3PWMH, 289
D3PWMT, 289
D3Status, 289
DDHH, 289
DELAYT, 289
FILTK, 289
GPSBAUD, 289
GPSHeight, 289
GPSTYPE, 289
GPSVH, 289
GPSVL, 290
GPSVX_H, 290
GPSVX_L, 290
GPSVY_H, 290
GPSVY_L, 290
GPSVZ_H, 290
GPSVZ_L, 290
GPSYAW, 290
GX, 290
GXOFFSET, 290
GY, 291
GYOFFSET, 291
GYROCALITHR, 291
GYROCALTIME, 291
GYROFILT, 291
GYRORANGE, 291
GYROSENSOR, 291
GYROZSCALE, 291
GZ, 291
GZOFFSET, 291
HDOP, 292
HeightH, 292
HeightL, 292
HPitch, 292
HRoll, 292
HXi, 292
HXOFFSET, 292
HYaw, 292
HYi, 292
HYOFFSET, 292
HZi, 293
HZOFFSET, 293
IICADDR, 293
KEY, 293
KEY_UNLOCK, 293
LatH, 293
LatL, 293
LEDOFF, 293
LonH, 293
LonL, 293
LPitch, 294
LRoll, 294
LYaw, 294
MAGFILT, 294
MAGRANGX, 294
MAGRANGY, 294
MAGRANGZ, 294
MAGSENSOR, 294
MMSS, 294
MODDELAY, 294
MOVESTA, 295
MOVETHR, 295
MS, 295
NORMAL, 295
NUMBERID1, 295
NUMBERID2, 295
NUMBERID3, 295
NUMBERID4, 295
NUMBERID5, 295
NUMBERID6, 295
ORIENT, 296
ORIENT_HERIZONE, 296
ORIENT_VERTICLE, 296
PDOP, 296
Pitch, 296
POWONSEND, 296

PRESSENSOR, 296
PressureH, 296
PressureL, 296
q0, 296
q1, 297
q2, 297
q3, 297
READADDR, 297
REFPITCH, 297
REFROLL, 297
REFYAW, 297
REGINITFLAG, 297
REGSIZE, 297
Roll, 297
RRATE, 298
RRATE_02HZ, 298
RRATE_05HZ, 298
RRATE_100HZ, 298
RRATE_10HZ, 298
RRATE_125HZ, 298
RRATE_1HZ, 298
RRATE_200HZ, 298
RRATE_20HZ, 298
RRATE_2HZ, 298
RRATE_50HZ, 299
RRATE_5HZ, 299
RRATE_NONE, 299
RRATE_ONCE, 299
RSW, 299
RSW_ACC, 299
RSW_ANGLE, 299
RSW_GPS, 299
RSW_GSA, 299
RSW_GYRO, 299
RSW_MAG, 300
RSW_MASK, 300
RSW_PORT, 300
RSW_PRESS, 300
RSW_Q, 300
RSW_TIME, 300
RSW_V, 300
SAVE, 300
SAVE_PARAM, 300
SAVE_SWRST, 300
SLEEP, 301
SVNUM, 301
TEMP, 301
TEMP905x, 301
TIMEZONE, 301
TRIGTIME, 301
VDOP, 301
VERSION, 301
WERROR, 301
WIT_ACC, 301
WIT_ANGLE, 302
WIT_BAUD_115200, 302
WIT_BAUD_19200, 302
WIT_BAUD_230400, 302
WIT_BAUD_38400, 302
WIT_BAUD_460800, 302
WIT_BAUD_4800, 302
WIT_BAUD_57600, 302
WIT_BAUD_921600, 302
WIT_BAUD_9600, 302
WIT_DPORT, 303
WIT_GPS, 303
WIT_GSA, 303
WIT_GYRO, 303
WIT_MAGNETIC, 303
WIT_PRESS, 303
WIT_QUATER, 303
WIT_REGVALUE, 303
WIT_TIME, 303
WIT_VELOCITY, 303
WZCNT, 304
WZSTATIC, 304
WZTIME, 304
XA30NSCALE, 304
XA30PSCALE, 304
XA85NSCALE, 304
XA85PSCALE, 304
XMAX, 304
XMIN, 304
XREFROLL, 304
XRSCALE, 305
YA30NSCALE, 305
YA30PSCALE, 305
YA85NSCALE, 305
YA85PSCALE, 305
Yaw, 305
YMAX, 305
YMIN, 305
YREFPITCH, 305
YRSCALE, 305
YYMM, 306
ZREFYAW, 306
ZRSCALE, 306
REGINITFLAG
 REG.h, 297
REGSIZE
 REG.h, 297
RegUpdateCb
 wit_c_sdk.h, 226
reinit
 WidgetAI, 151
 WidgetALT, 153
 WidgetASI, 155
 WidgetEADI, 158
 WidgetEHSI, 163
 WidgetHI, 166
 WidgetTC, 171
 WidgetVSI, 173
relative
 AltimeterData, 23
RemoteSelector, 135
 ~RemoteSelector, 136

RemoteSelector, 136
 service, 137
 startDiscovery, 137
 stopDiscovery, 137
 replaceItem
 LayoutSquare, 71
 RESET
 MicroQiskitCpp.h, 260
 ret_imu
 MyTcpSocket, 122
 ret_transponder
 MyTcpSocket, 122
 Roll
 REG.h, 297
 roll_blended
 MainWindow, 91
 roll_blended_ok
 MainWindow, 91
 rotateSensors
 rotation_matrix.cpp, 323, 324
 rotation_matrix.h, 216
 rotateVector
 rotation_matrix.cpp, 324
 rotation_matrix.h, 216
 rotation_matrix.cpp
 createRotationMatrix, 322
 createRotationMatrixZ, 323
 multiplyMatrix, 323
 printVector, 323
 rotateSensors, 323, 324
 rotateVector, 324
 rotation_matrix.h
 createRotationMatrix, 215
 createRotationMatrixZ, 215
 DEG_TO_RAD, 213
 M_PI, 213
 Matrix3x3, 214
 Matrix3x6, 214
 Matrix9x9, 214
 multiplyMatrix, 215
 printVector, 215
 RAD_TO_DEG, 214
 rotateSensors, 216
 rotateVector, 216
 Vector3x, 214
 Vector9, 214
 rotationMatrix
 MainWindow, 91
 rotfilterlength
 mainwindow.cpp, 355
 rPos
 MyTcpSocket, 123
 RRATE
 REG.h, 298
 RRATE_02HZ
 REG.h, 298
 RRATE_05HZ
 REG.h, 298
 RRATE_100HZ
 REG.h, 298
 RRATE_10HZ
 REG.h, 298
 RRATE_125HZ
 REG.h, 298
 RRATE_1HZ
 REG.h, 298
 RRATE_200HZ
 REG.h, 298
 RRATE_20HZ
 REG.h, 298
 RRATE_2HZ
 REG.h, 298
 RRATE_50HZ
 REG.h, 299
 RRATE_5HZ
 REG.h, 299
 RRATE_NONE
 REG.h, 299
 RRATE_ONCE
 REG.h, 299
 rSpeed
 MyTcpSocket, 123
 RSW
 REG.h, 299
 RSW_ACC
 REG.h, 299
 RSW_ANGLE
 REG.h, 299
 RSW_GPS
 REG.h, 299
 RSW_GSA
 REG.h, 299
 RSW_GYRO
 REG.h, 299
 RSW_MAG
 REG.h, 300
 RSW_MASK
 REG.h, 300
 RSW_PORT
 REG.h, 300
 RSW_PRESS
 REG.h, 300
 RSW_Q
 REG.h, 300
 RSW_TIME
 REG.h, 300
 RSW_V
 REG.h, 300
 rx
 QuantumCircuit, 131
 RxCallback
 BleUart, 25
 BTComQt, 28
 ComBt, 44
 ComQt, 49
 RxCallbackINS

ins_driver.cpp, 347
ry QuantumCircuit, 131
rz QuantumCircuit, 132

s_cDataUpdate ins_driver.cpp, 350
s_ucAddr wit_c_sdk.c, 341
s_ucWitDataBuff wit_c_sdk.c, 341
s_uiProtocolo wit_c_sdk.c, 341
s_uiReadRegIndex wit_c_sdk.c, 342
s_uiWitDataCnt wit_c_sdk.c, 342
SAVE REG.h, 300
SAVE_PARAM REG.h, 300
SAVE_SWRST REG.h, 300
SCALE mainwindow.cpp, 355
scancomplete BleUart, 27
scanDone ComBt, 47
SCANF_INT wit_c_sdk.h, 223
scanServices Device, 59
SCREEN mytcpsocket.h, 271
screen_index MainWindow, 91
ScreenMode MainWindow, 91
send BTComQt, 29, 30
ComBt, 46
ComQt, 52, 53
SendData Device, 59
sendData TcpClient, 148
sendMessage ChatClient, 37
ChatServer, 40
MqttClient, 94
MyTcpSocket, 114
SENSOR ssdp.h, 315
SENSOR_TYPE ssdp.h, 315
SensorService, 137 name, 137

st, 137
uuid, 137
SensorUartSend ins_driver.cpp, 349
serial PortEntry, 127
serial_ BTComQt, 30
ComBt, 47
ComQt, 54
serialport MyTcpSocket, 123
serialport_android.cpp
getSelf, 326
Java_com_hoho_android_usbserial_driver_TestClassTerje_nativeOn
327
Java_com_hoho_android_usbserial_driver_TestClassTerje_nativeOn
327
Java_com_hoho_android_usbserial_driver_TestClassTerje_nativeOn
328
serialPortb ins_driver.cpp, 351
serialPorts ins_driver.cpp, 351
SerialWrite wit_c_sdk.h, 226
series MainWindow, 91
SERVER_ADDRESS MyTcpSocket, 123
service RemoteSelector, 137
ServiceInfo, 139
serviceChanged ServiceInfo, 140
ServiceInfo, 138
getName, 139
getType, 139
getUuid, 139
service, 139
serviceChanged, 140
ServiceInfo, 139
serviceUuid chatserver.cpp, 344
set_default_config MainWindow, 83
set_default_planes MainWindow, 84
set_default_radio MainWindow, 84
set_registers QuantumCircuit, 133
setAirspeed WidgetASI, 155
WidgetEADI, 158
setAirspeedSel WidgetEADI, 158
setAltitude

WidgetALT, 153
 WidgetEADI, 158
 setAltitudeSel
 WidgetEADI, 158
 setbacklit
 MyTcpSocket, 114
 setBaudrate
 ComBt, 46
 ComQt, 53
 setBearing
 WidgetEHSI, 163
 setButtonIcon
 MainWindow, 84
 setCamera
 MainWindow, 85
 setCharacteristic
 CharacteristicInfo, 34
 setClimbRate
 WidgetEADI, 158
 WidgetVSI, 173
 setCourse
 WidgetEHSI, 164
 setDeviation
 WidgetEHSI, 164
 setDevice
 DeviceInfo, 64
 setDistance
 WidgetEHSI, 164
 setDots
 WidgetEADI, 158
 setFD
 WidgetEADI, 159
 setFltMode
 WidgetEADI, 159
 setFPM
 WidgetEADI, 159
 setGeometry
 LayoutSquare, 72
 setHeading
 WidgetEADI, 159
 WidgetEHSI, 164
 WidgetHI, 166
 setHeadingSel
 WidgetEADI, 159
 WidgetEHSI, 164
 setIMU
 MainWindow, 85
 setINSRxCallback
 ins_driver.cpp, 350
 setIosParams
 myNativeWrapperFunctions.h, 269
 setLNAV
 WidgetEADI, 160
 setMachNo
 WidgetEADI, 160
 setMessageHandler
 MqttClient, 95
 setmode

 MainWindow, 85
 setPitch
 WidgetAI, 151
 WidgetEADI, 160
 setPressure
 WidgetALT, 153
 WidgetEADI, 160
 setQNH
 MainWindow, 86
 setRoll
 WidgetAI, 151
 WidgetEADI, 160
 setRxCallback
 BleUart, 26
 BTComQt, 30
 ComBt, 46
 ComQt, 54
 setSerialPorts
 MyTcpSocket, 114
 setSlipSkid
 WidgetEADI, 160
 WidgetTC, 171
 setSpdMode
 WidgetEADI, 160
 setStall
 WidgetEADI, 160
 setText
 NoButtonMessageBox, 125
 setTurnRate
 WidgetEADI, 161
 WidgetTC, 171
 setVal
 Device, 60
 setVfe
 WidgetEADI, 161
 setVNAV
 WidgetEADI, 161
 setVne
 WidgetEADI, 161
 shots
 Simulator, 143
 showImage
 MainWindow, 86
 simGPS
 mytcpsocket.h, 271
 Simulator, 140
 get_counts, 141
 get_memory, 141
 get_qasm, 142
 get_qiskit, 142
 get_statevector, 142
 qc, 143
 shots, 143
 Simulator, 141
 sizeHint
 LayoutSquare, 72
 SLEEP
 REG.h, 301

SLEEP_MS
 wit_c_sdk.h, 223

socketErrorOccurred
 ChatClient, 37

speed
 TrackPoint, 149

splash
 MainWindow, 92

sport
 MyTcpSocket, 123

sReg
 wit_c_sdk.c, 342
 wit_c_sdk.h, 234

ssdp.h
 SENSOR, 315
 SENSOR_TYPE, 315
 SSDP_GROUP, 315
 SSDP_PORT, 315
 SSDP_ST, 315

SSDP_GROUP
 ssdp.h, 315

SSDP_PORT
 ssdp.h, 315

SSDP_ST
 ssdp.h, 315

ssdpConfig
 MyTcpSocket, 115

SsdpDiscoverer, 143
 deviceFound, 145
 SsdpDiscoverer, 144
 startDiscovery, 145

st
 SensorService, 137

start
 MulticastListener, 97

startClient
 ChatClient, 37

startDeviceDiscovery
 Device, 60

startDiscovery
 RemoteSelector, 137
 SsdpDiscoverer, 145

startScan
 BleUart, 26

startServer
 ChatServer, 41

state
 Device, 60

stateChanged
 Device, 60

status
 ComBt, 47
 ComQt, 54

stop
 MulticastListener, 98

stopClient
 ChatClient, 37

stopDiscovery

 RemoteSelector, 137

stopServer
 ChatServer, 41

STX
 mytcpsocket.h, 272

subscribe
 MqttClient, 95

SVNUM
 REG.h, 301

systemLocation
 PortEntry, 127

take
 LayoutSquare, 73

takeAt
 LayoutSquare, 73

takeoff_altitude
 MainWindow, 92

takeoff_latitude
 MainWindow, 92

takeoff_longitude
 MainWindow, 92

TcpClient, 146
 connected, 147
 connectTo, 147
 dataReceived, 147
 disconnected, 147
 disconnectFrom, 147
 errorOccurred, 147
 sendData, 148
 TcpClient, 147

TEMP
 REG.h, 301

Temp
 MyTcpSocket, 123

TEMP905x
 REG.h, 301

temperature
 AltimeterData, 23

test
 Device, 61

text
 MyTcpSocket, 123

time
 TrackPoint, 149

timerAndroid
 BTComQt, 30

timerEvent
 MainWindow, 86

TIMEZONE
 REG.h, 301

TrackPoint, 148
 dt, 148
 elevation, 148
 latitude, 148
 longitude, 148
 speed, 149
 time, 149

TRANSPONDERLOG

mainwindow.h, 252
 Transponderstat
 MyTcpSocket, 123
 TransponderstatWithBarometer
 MyTcpSocket, 124
 TRIGTIME
 REG.h, 301
 UDP
 mytcpsocket.cpp, 359
 Ui, 21
 ui
 MainWindow, 92
 UINT32
 wit_c_sdk.h, 224
 updateCameras
 MainWindow, 86
 updateChanged
 Device, 61
 updateDisplay
 Device, 61
 updateValueSet
 Device, 61
 USE_BT_IMU
 mytcpsocket.h, 271
 use_ins_only
 MyTcpSocket, 124
 USE_KeepAwakeHelper
 mainwindow.h, 252
 uuid
 SensorService, 137
 valid
 GeoidHelper::Impl, 67
 varfilterlength
 mainwindow.cpp, 356
 VDOP
 REG.h, 301
 Vector3x
 rotation_matrix.h, 214
 Vector9
 rotation_matrix.h, 214
 vendorId
 PortEntry, 127
 VER
 MyTcpSocket, 124
 VERSION
 REG.h, 301
 view
 MainWindow, 92
 WERROR
 REG.h, 301
 WidgetAI, 149
 ~WidgetAI, 150
 redraw, 151
 reinit, 151
 setPitch, 151
 setRoll, 151
 WidgetAI, 150
 WidgetALT, 151
 ~WidgetALT, 153
 redraw, 153
 reinit, 153
 setAltitude, 153
 setPressure, 153
 WidgetALT, 152
 WidgetASI, 154
 ~WidgetASI, 155
 redraw, 155
 reinit, 155
 setAirspeed, 155
 WidgetASI, 154
 WidgetEADI, 156
 ~WidgetEADI, 157
 redraw, 158
 reinit, 158
 setAirspeed, 158
 setAirspeedSel, 158
 setAltitude, 158
 setAltitudeSel, 158
 setClimbRate, 158
 setDots, 158
 setFD, 159
 setFltMode, 159
 setFPM, 159
 setHeading, 159
 setHeadingSel, 159
 setLNAV, 160
 setMachNo, 160
 setPitch, 160
 setPressure, 160
 setRoll, 160
 setSlipSkid, 160
 setSpdMode, 160
 setStall, 160
 setTurnRate, 161
 setVfe, 161
 setVNAV, 161
 setVne, 161
 WidgetEADI, 157
 WidgetEHSI, 162
 ~WidgetEHSI, 163
 redraw, 163
 reinit, 163
 setBearing, 163
 setCourse, 164
 setDeviation, 164
 setDistance, 164
 setHeading, 164
 setHeadingSel, 164
 WidgetEHSI, 163
 WidgetHI, 165
 ~WidgetHI, 166
 redraw, 166
 reinit, 166
 setHeading, 166

WidgetHI, 165
WidgetSix, 167
 ~WidgetSix, 168
 getAI, 168
 getALT, 168
 getASI, 168
 getEADI, 169
 getHI, 169
 getTC, 169
 getVSI, 169
 WidgetSix, 168
WidgetTC, 169
 ~WidgetTC, 171
 redraw, 171
 reinit, 171
 setSlipSkid, 171
 setTurnRate, 171
 WidgetTC, 170
WidgetVSI, 172
 ~WidgetVSI, 173
 redraw, 173
 reinit, 173
 setClimbRate, 173
 WidgetVSI, 172
WIT_ACC
 REG.h, 301
WIT_ANGLE
 REG.h, 302
WIT_BAUD_115200
 REG.h, 302
WIT_BAUD_19200
 REG.h, 302
WIT_BAUD_230400
 REG.h, 302
WIT_BAUD_38400
 REG.h, 302
WIT_BAUD_460800
 REG.h, 302
WIT_BAUD_4800
 REG.h, 302
WIT_BAUD_57600
 REG.h, 302
WIT_BAUD_921600
 REG.h, 302
WIT_BAUD_9600
 REG.h, 302
wit_c_sdk.c
 __CRC16, 331
 __CalSum, 331
 __ auchCRCHi, 340
 __ auchCRCLo, 341
 CheckRange, 331
 CopeWitData, 331
 FuncR, 330
 FuncW, 330
 p_WitDelaymsFunc, 341
 p_WitRegUpdateCbFunc, 341
 p_WitSerialWriteFunc, 341
posix_delay_ms, 332
s_ucAddr, 341
s_ucWitDataBuff, 341
s_uiProtocolo, 341
s_uiReadRegIndex, 342
s_uiWitDataCnt, 342
sReg, 342
WitCaliRefAngle, 332
WitDeInit, 332
WitDelayMsRegister, 333
WitInit, 333
WitReadReg, 333
WitRegisterCallBack, 333
WitSaveParameter, 334
WitSerialDataIn, 334
WitSerialWriteRegister, 335
WitSetBandwidth, 335
WitSetCanBaud, 335
WitSetContent, 336
WitSetForReset, 336
WitSetOutputRate, 336
WitSetUartBaud, 337
WitStartAccCali, 337
WitStartMagCali, 338
WitStopAccCali, 338
WitStopMagCali, 338
WitWriteReg, 339
wit_c_sdk.h
 CheckRange, 226
 DelaymsCb, 226
 posix_delay_ms, 226
 RegUpdateCb, 226
 SCANF_INT, 223
 SerialWrite, 226
 SLEEP_MS, 223
 sReg, 234
 UINT32, 224
 WIT_DATA_BUFF_SIZE, 224
 WIT_HAL_BUSY, 224
 WIT_HAL_EMPTY, 224
 WIT_HAL_ERROR, 224
 WIT_HAL_INVAL, 224
 WIT_HAL_NOMEM, 224
 WIT_HAL_OK, 224
 WIT_HAL_TIMEOUT, 225
 WIT_PROTOCOL_905x_CAN, 225
 WIT_PROTOCOL_905x_MODBUS, 225
 WIT_PROTOCOL_CAN, 225
 WIT_PROTOCOL_I2C, 225
 WIT_PROTOCOL_JY61, 225
 WIT_PROTOCOL_MODBUS, 225
 WIT_PROTOCOL_NORMAL, 225
 WitCaliRefAngle, 227
 WitDeInit, 227
 WitDelayMsRegister, 227
 WitInit, 227
 WitReadReg, 228
 WitRegisterCallBack, 228

WitSaveParameter, 228
WitSerialDataIn, 229
WitSerialWriteRegister, 229
WitSetBandwidth, 230
WitSetCanBaud, 230
WitSetContent, 230
WitSetForReset, 231
WitSetOutputRate, 231
WitSetUartBaud, 231
WitStartAccCali, 232
WitStartMagCali, 232
WitStopAccCali, 233
WitStopMagCali, 233
WitWriteReg, 233
WIT_DATA_BUFF_SIZE
 wit_c_sdk.h, 224
WIT_DPRT
 REG.h, 303
WIT_GPS
 REG.h, 303
WIT_GSA
 REG.h, 303
WIT_GYRO
 REG.h, 303
WIT_HAL_BUSY
 wit_c_sdk.h, 224
WIT_HAL_EMPTY
 wit_c_sdk.h, 224
WIT_HAL_ERROR
 wit_c_sdk.h, 224
WIT_HAL_INVAL
 wit_c_sdk.h, 224
WIT_HAL_NOMEM
 wit_c_sdk.h, 224
WIT_HAL_OK
 wit_c_sdk.h, 224
WIT_HAL_TIMEOUT
 wit_c_sdk.h, 225
WIT_MAGNETIC
 REG.h, 303
WIT_PRESS
 REG.h, 303
WIT_PROTOCOL_905x_CAN
 wit_c_sdk.h, 225
WIT_PROTOCOL_905x_MODBUS
 wit_c_sdk.h, 225
WIT_PROTOCOL_CAN
 wit_c_sdk.h, 225
WIT_PROTOCOL_I2C
 wit_c_sdk.h, 225
WIT_PROTOCOL_JY61
 wit_c_sdk.h, 225
WIT_PROTOCOL_MODBUS
 wit_c_sdk.h, 225
WIT_PROTOCOL_NORMAL
 wit_c_sdk.h, 225
WIT_QUATER
 REG.h, 303

WIT_REGVALUE
 REG.h, 303
WIT_TIME
 REG.h, 303
WIT_VELOCITY
 REG.h, 303
WitCaliRefAngle
 wit_c_sdk.c, 332
 wit_c_sdk.h, 227
WitDeInit
 wit_c_sdk.c, 332
 wit_c_sdk.h, 227
WitDelayMsRegister
 wit_c_sdk.c, 333
 wit_c_sdk.h, 227
WitInit
 wit_c_sdk.c, 333
 wit_c_sdk.h, 227
WitReadReg
 wit_c_sdk.c, 333
 wit_c_sdk.h, 228
WitRegisterCallBack
 wit_c_sdk.c, 333
 wit_c_sdk.h, 228
WitSaveParameter
 wit_c_sdk.c, 334
 wit_c_sdk.h, 228
WitSerialDataIn
 wit_c_sdk.c, 334
 wit_c_sdk.h, 229
WitSerialWriteRegister
 wit_c_sdk.c, 335
 wit_c_sdk.h, 229
WitSetBandwidth
 wit_c_sdk.c, 335
 wit_c_sdk.h, 230
WitSetCanBaud
 wit_c_sdk.c, 335
 wit_c_sdk.h, 230
WitSetContent
 wit_c_sdk.c, 336
 wit_c_sdk.h, 230
WitSetForReset
 wit_c_sdk.c, 336
 wit_c_sdk.h, 231
WitSetOutputRate
 wit_c_sdk.c, 336
 wit_c_sdk.h, 231
WitSetUartBaud
 wit_c_sdk.c, 337
 wit_c_sdk.h, 231
WitStartAccCali
 wit_c_sdk.c, 337
 wit_c_sdk.h, 232
WitStartMagCali
 wit_c_sdk.c, 338
 wit_c_sdk.h, 232
WitStopAccCali

wit_c_sdk.c, 338
wit_c_sdk.h, 233
WitStopMagCali
 wit_c_sdk.c, 338
 wit_c_sdk.h, 233
WitWriteReg
 wit_c_sdk.c, 339
 wit_c_sdk.h, 233
writeBytes
 BleUart, 26
wt901Rx
 bleuart.h, 203
wt901Service
 bleuart.h, 203
wt901Tx
 bleuart.h, 203
WZCNT
 REG.h, 304
WZSTATIC
 REG.h, 304
WZTIME
 REG.h, 304

x
 QuantumCircuit, 133
XA30NSCALE
 REG.h, 304
XA30PSCALE
 REG.h, 304
XA85NSCALE
 REG.h, 304
XA85PSCALE
 REG.h, 304
XMAX
 REG.h, 304
XMIN
 REG.h, 304
XREFROLL
 REG.h, 304
XRSCALE
 REG.h, 305

y
 QuantumCircuit, 133
YA30NSCALE
 REG.h, 305
YA30PSCALE
 REG.h, 305
YA85NSCALE
 REG.h, 305
YA85PSCALE
 REG.h, 305
Yaw
 REG.h, 305
YMAX
 REG.h, 305
YMIN
 REG.h, 305
YREFPITCH
 REG.h, 305
 YRSCALE
 REG.h, 305
 YYMM
 REG.h, 306

z

QuantumCircuit, 134

ZREFYAW
 REG.h, 306ZRSCALE
 REG.h, 306