QMAST Source Code

Generated by Doxygen 1.7.5.1

Fri Sep 30 2011 22:43:52

Contents

1 The QMAST Alpha 6 Sailing Code 1

2 Module Index 3

2.1 Modules . . . 3

3 Data Structure Index 5

3.1 Data Structures . . . 5

4 File Index 7

4.1 File List . . . 7

5 Module Documentation 9

5.1 Globalconstants . . . 9

5.1.1 Detailed Description . . . 10

5.1.2 Define Documentation . . . 10

5.1.2.1 ALL_IN . . . 10

5.1.2.2 ALL_OUT . . . 10

5.1.2.3 badCompassDataBit . . . 11

5.1.2.4 badGpsData . . . 11

5.1.2.5 badWindData . . . 11

5.1.2.6 BUFF_MAX . . . 11

5.1.2.7 checksumBadBit . . . 11

5.1.2.8 DEGREE_TO_MINUTE . . . 11

5.1.2.9 LATITUDE_TO_METER . . . 11

5.1.2.10 LONGEST_NMEA . . . 12

5.1.2.11 LONGITUDE_TO_METER . . . 12

5.1.2.12 MARK_DISTANCE . . . 12

5.1.2.13 noDataBit . . . 12
5.1.2.14 oldDataBit . . . 12
5.1.2.15 resetPin . . . 12
5.1.2.16 rolloverDataBit . . . 12
5.1.2.17 SHORTEST_NMEA . . . 12
5.1.2.18 STATION_KEEPING_RADIUS . . . 13
5.1.2.19 TACKING_ANGLE . . . 13
5.1.2.20 tooMuchRollBit . . . 13
5.1.2.21 twoCommasBit . . . 13
5.1.2.22 txPin . . . 13
5.1.2.23 WIND_CHANGE_THRESHOLD . . . 13
5.2 SerialData . . . 14
5.2.1 Detailed Description . . . 15
5.2.2 Variable Documentation . . . 16
5.2.2.1 bspeed . . . 16
5.2.2.2 bspeedk . . . 16
5.2.2.3 currentPoint . . . 16
5.2.2.4 CurrentSelection . . . 16
5.2.2.5 deviation . . . 16
5.2.2.6 distanceVal . . . 16
5.2.2.7 errorCode . . . 16
5.2.2.8 extraCompassData . . . 16
5.2.2.9 extraCompassDataArray . . . 17
5.2.2.10 extraWindData . . . 17
5.2.2.11 extraWindDataArray . . . 17
5.2.2.12 heading . . . 17
5.2.2.13 heading_newest . . . 17
5.2.2.14 headingc . . . 17
5.2.2.15 headingVal . . . 17
5.2.2.16 ironTime . . . 17
5.2.2.17 jibVal . . . 18
5.2.2.18 mainVal . . . 18
5.2.2.19 pitch . . . 18
5.2.2.20 point . . . 18

5.2.2.21 points . . . 18
5.2.2.22 roll . . . 18
5.2.2.23 rudderDir . . . 18
5.2.2.24 rudderVal . . . 18
5.2.2.25 savedCompassChecksum . . . 19
5.2.2.26 savedCompassXorState . . . 19
5.2.2.27 savedWindChecksum . . . 19
5.2.2.28 savedWindXorState . . . 19
5.2.2.29 servo_ser . . . 19
5.2.2.30 startTime . . . 19
5.2.2.31 stationCounter . . . 19
5.2.2.32 StationKeepingTimeInBox . . . 19
5.2.2.33 StraightSailDirection . . . 19
5.2.2.34 tacking . . . 20
5.2.2.35 tackingSide . . . 20
5.2.2.36 timesUp . . . 20
5.2.2.37 trueWind . . . 20
5.2.2.38 variance . . . 20
5.2.2.39 wind_angl . . . 20
5.2.2.40 wind_angl_newest . . . 20
5.2.2.41 wind_velocity . . . 20

6 Data Structure Documentation **21**

6.1 points Struct Reference . . . 21
6.1.1 Detailed Description . . . 21
6.1.2 Field Documentation . . . 21
6.1.2.1 latDeg . . . 21
6.1.2.2 latMin . . . 21
6.1.2.3 lonDeg . . . 21
6.1.2.4 lonMin . . . 22

7 File Documentation **23**

7.1 /Users/allgood38/Desktop/qmast/sailcode_alpha6/DataAcquisition.pde File Reference . . . 23
7.1.1 Function Documentation . . . 23

7.1.1.1 checkErrorBit . . . 23
7.1.1.2 clearErrorBit . . . 23
7.1.1.3 sensorData . . . 24
7.1.1.4 setErrorBit . . . 25
7.2 /Users/allgood38/Desktop/qmast/sailcode_alpha6/LocationStruct.h File Reference . . . 26
7.2.1 Variable Documentation . . . 26
7.2.1.1 boatLocation . . . 26
7.2.1.2 clearPoints . . . 26
7.2.1.3 coursePoints . . . 27
7.2.1.4 floatingStationPoints . . . 27
7.2.1.5 stationPoints . . . 27
7.2.1.6 stayPoint . . . 27
7.2.1.7 waypoints . . . 27
7.3 /Users/allgood38/Desktop/qmast/sailcode_alpha6/Menu.pde File Reference . . . 27
7.3.1 Function Documentation . . . 28
7.3.1.1 displayMenu . . . 28
7.4 /Users/allgood38/Desktop/qmast/sailcode_alpha6/MotorControl-Functions.pde File Reference . . . 28
7.4.1 Function Documentation . . . 29
7.4.1.1 servo_command . . . 29
7.4.1.2 setJib . . . 29
7.4.1.3 setMain . . . 30
7.4.1.4 setrudder . . . 31
7.4.1.5 setSails . . . 32
7.5 /Users/allgood38/Desktop/qmast/sailcode_alpha6/NavigationCode.pde File Reference . . . 33
7.5.1 Function Documentation . . . 33
7.5.1.1 getCloseHauledDirn . . . 33
7.5.1.2 getOppositeCloseHauledDirn . . . 34
7.5.1.3 getWaypointDirn . . . 34
7.5.1.4 getWindDirn . . . 35
7.5.1.5 GPSdistance . . . 36

7.6 /Users/allgood38/Desktop/qmast/sailcode_alpha6/oldtack.pde File Reference . . . 36
7.6.1 Function Documentation . . . 36
7.6.1.1 getOutofIronsOld . . . 36
7.6.1.2 oldtack . . . 37
7.6.1.3 oldtack2 . . . 38
7.7 /Users/allgood38/Desktop/qmast/sailcode_alpha6/ParsingFunctions.pde File Reference . . . 39
7.7.1 Function Documentation . . . 39
7.7.1.1 DataValid . . . 39
7.7.1.2 ParseGPGLL . . . 40
7.7.1.3 Parser . . . 40
7.8 /Users/allgood38/Desktop/qmast/sailcode_alpha6/sailcode_alpha6.pde File Reference . . . 41
7.8.1 Function Documentation . . . 45
7.8.1.1 loop . . . 45
7.8.1.2 setup . . . 46
7.9 /Users/allgood38/Desktop/qmast/sailcode_alpha6/SailingLogic.pde File Reference . . . 47
7.9.1 Function Documentation . . . 47
7.9.1.1 checkTack . . . 47
7.9.1.2 rudderControl . . . 48
7.9.1.3 sail . . . 49
7.9.1.4 sailControl . . . 51
7.9.1.5 sailCourse . . . 51
7.9.1.6 sailToWaypoint . . . 52
7.10 /Users/allgood38/Desktop/qmast/sailcode_alpha6/StationKeeping.pde File Reference . . . 54
7.10.1 Function Documentation . . . 54
7.10.1.1 fillStationKeepingWaypoints . . . 54
7.10.1.2 getStationKeepingCentre . . . 54
7.10.1.3 stationKeep . . . 55
7.10.1.4 stationKeepSinglePoint . . . 55
7.11 /Users/allgood38/Desktop/qmast/sailcode_alpha6/Tack.pde File Reference . . . 56
7.11.1 Function Documentation . . . 56

7.11.1.1 getOutofIrons 56
7.11.1.2 tack 57
7.12 /Users/allgood38/Desktop/qmast/sailcode_alpha6/Testing_Functions.pde File Reference 58
7.13 /Users/allgood38/Desktop/qmast/sailcode_alpha6/Transmit.pde File - Reference 58
7.13.1 Function Documentation 59
7.13.1.1 relayData 59
7.13.1.2 transmit 59
7.14 /Users/allgood38/Desktop/qmast/sailcode_alpha6/Utilities.pde File - Reference 59
7.14.1 Function Documentation 60
7.14.1.1 between 60
7.14.1.2 convertASCIItoHex 60
7.14.1.3 degreesToRadians 61
7.14.1.4 radiansToDegrees 61

Chapter 1

The QMAST Alpha 6 Sailing Code

This Documentation

Is the same style as the JavaDoc documentation. The same commands are used in the code, and generates these pages. For info on the program used check out a program called Doxygen

Revised by Laszlo 2011-05-13

Ported to Arudino November 2010 by Christine and the supercool software team Created on: 2010-05-11 Author: Nader for MAST Software

Chapter 2

Module Index

2.1 Modules

Here is a list of all modules:

Globalconstants 9
SerialData 14

Chapter 3

Data Structure Index

3.1 Data Structures

Here are the data structures with brief descriptions:

points 21

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

/Users/allgood38/Desktop/qmast/sailcode_alpha6/DataAcquisition.pde 23
/Users/allgood38/Desktop/qmast/sailcode_alpha6/LocationStruct.h 26
/Users/allgood38/Desktop/qmast/sailcode_alpha6/Menu.pde 27
/Users/allgood38/Desktop/qmast/sailcode_alpha6/MotorControlFunctions.pde 28
/Users/allgood38/Desktop/qmast/sailcode_alpha6/NavigationCode.pde 33
/Users/allgood38/Desktop/qmast/sailcode_alpha6/oldtack.pde 36
/Users/allgood38/Desktop/qmast/sailcode_alpha6/ParsingFunctions.pde . . . 39
/Users/allgood38/Desktop/qmast/sailcode_alpha6/sailcode_alpha6.pde . . . 41
/Users/allgood38/Desktop/qmast/sailcode_alpha6/SailingLogic.pde 47
/Users/allgood38/Desktop/qmast/sailcode_alpha6/StationKeeping.pde 54
/Users/allgood38/Desktop/qmast/sailcode_alpha6/Tack.pde 56
/Users/allgood38/Desktop/qmast/sailcode_alpha6/Testing_Functions.pde . . . 58
/Users/allgood38/Desktop/qmast/sailcode_alpha6/Transmit.pde 58
/Users/allgood38/Desktop/qmast/sailcode_alpha6/Utilities.pde 59

Chapter 5

Module Documentation

5.1 Globalconstants

for pololu non-buffering serial channel

Defines

- #define TACKING_ANGLE 40

  Boat parameter constants in globalconstants.
- #define MARK_DISTANCE 4

  Course Navigation constants.
- #define STATION_KEEPING_RADIUS 15

  Station keeping navigation constants.
- #define WIND_CHANGE_THRESHOLD 10
- #define BUFF_MAX 511

  serial data constants
- #define DEGREE_TO_MINUTE 60

  Calculation constantes.
- #define LATITUDE_TO_METER 1855

  There are (approximately) 1855 meters in a minute of latitude everywhere; This isn't true for longitude, as it depends on the latitude.
- #define LONGITUDE_TO_METER 1314
- #define noDataBit 0

  no data, error error bit
- #define oldDataBit 1

  there is data, but buffer is full, error bit
- #define checksumBadBit 2

  indicates checksum fail on data
- #define twoCommasBit 3

indicates that there were two commas in the data, and it has been discarded and not parsed

- #define rolloverDataBit 4

  Indicates data rolled over, not fast enough.
- #define badCompassDataBit 5

  indicates that strtok did not return PTNTHTM, so we probably got bad data
- #define tooMuchRollBit 6

  indicates the boat is falling over
- #define badWindData 7

  indicates an error from the wind sensor
- #define badGpsData 8

  indicates error in gps data
- #define ALL_IN 0

  Constant which defines when the boat is "All In".
- #define ALL_OUT 100

  Constant which defines when the boat is "All Out".
- #define resetPin 8

  Pololu reset (digital pin on arduino)
- #define txPin 9

  Pololu serial pin (with SoftwareSerial library)
- #define SHORTEST_NMEA 5

  The shortest possible NMEA String.
- #define LONGEST_NMEA 120

  The longest possible NMEA String.

5.1.1 Detailed Description

for pololu non-buffering serial channel for parsing - necessary? Arduino doesn't look like it contains the String.h lib, however it does have its own string handling support built in. Global Constants

5.1.2 Define Documentation

5.1.2.1 #define ALL_IN 0

Constant which defines when the boat is "All In".

Definition at line 102 of file sailcode_alpha6.pde.

5.1.2.2 #define ALL_OUT 100

Constant which defines when the boat is "All Out".

Definition at line 104 of file sailcode_alpha6.pde.

5.1.2.3 #define badCompassDataBit 5

indicates that strtok did not return PTNTHTM, so we probably got bad data

Definition at line 92 of file sailcode_alpha6.pde.

5.1.2.4 #define badGpsData 8

indicates error in gps data

Definition at line 98 of file sailcode_alpha6.pde.

5.1.2.5 #define badWindData 7

indicates an error from the wind sensor

Definition at line 96 of file sailcode_alpha6.pde.

5.1.2.6 #define BUFF_MAX 511

serial data constants

Definition at line 61 of file sailcode_alpha6.pde.

5.1.2.7 #define checksumBadBit 2

indicates checksum fail on data

Definition at line 85 of file sailcode_alpha6.pde.

5.1.2.8 #define DEGREE_TO_MINUTE 60

Calculation constantes.

Definition at line 66 of file sailcode_alpha6.pde.

5.1.2.9 #define LATITUDE_TO_METER 1855

There are (approximately) 1855 meters in a minute of latitude everywhere; This isn't true for longitude, as it depends on the latitude.

There are approximately 1314 m in a minute of longitude at 45 degrees north (Kingston); this difference will mean that if we just use deltax over deltay in minutes to find an angle it will be wrong

there are 60 minutes in one degree

Definition at line 76 of file sailcode_alpha6.pde.

5.1.2.10 #define LONGEST_NMEA 120

The longest possible NMEA String.

Definition at line 116 of file sailcode_alpha6.pde.

5.1.2.11 #define LONGITUDE_TO_METER 1314

Definition at line 77 of file sailcode_alpha6.pde.

5.1.2.12 #define MARK_DISTANCE 4

Course Navigation constants.

Definition at line 48 of file sailcode_alpha6.pde.

5.1.2.13 #define noDataBit 0

no data, error error bit

Definition at line 81 of file sailcode_alpha6.pde.

5.1.2.14 #define oldDataBit 1

there is data, but buffer is full, error bit

Definition at line 83 of file sailcode_alpha6.pde.

5.1.2.15 #define resetPin 8

Pololu reset (digital pin on arduino)

Definition at line 108 of file sailcode_alpha6.pde.

5.1.2.16 #define rolloverDataBit 4

Indicates data rolled over, not fast enough.

Definition at line 90 of file sailcode_alpha6.pde.

5.1.2.17 #define SHORTEST_NMEA 5

The shortest possible NMEA String.

Definition at line 114 of file sailcode_alpha6.pde.

5.1.2.18 #define STATION_KEEPING_RADIUS 15

Station keeping navigation constants.

Definition at line 52 of file sailcode_alpha6.pde.

5.1.2.19 #define TACKING_ANGLE 40

Boat parameter constants in globalconstants.

Definition at line 45 of file sailcode_alpha6.pde.

5.1.2.20 #define tooMuchRollBit 6

indicates the boat is falling over

Definition at line 94 of file sailcode_alpha6.pde.

5.1.2.21 #define twoCommasBit 3

indicates that there were two commas in the data, and it has been discarded and not parsed

Definition at line 88 of file sailcode_alpha6.pde.

5.1.2.22 #define txPin 9

Pololu serial pin (with SoftwareSerial library)

Definition at line 110 of file sailcode_alpha6.pde.

5.1.2.23 #define WIND_CHANGE_THRESHOLD 10

Definition at line 56 of file sailcode_alpha6.pde.

5.2 SerialData

Variables for Serial Communications.

Variables

- int extraWindData = 0

  Contains extra data from the Wind Sensor.
- int extraCompassData = 0
- int savedWindChecksum = 0

  clear the global saved XOR value
- int savedWindXorState = 0

  clear the global saved XORstate value
- int savedCompassChecksum = 0
- int savedCompassXorState = 0
- char extraWindDataArray [LONGEST_NMEA]

  a buffer to store roll-over data in case this data is fetched mid-line
- char extraCompassDataArray [LONGEST_NMEA]
- float heading

  heading relative to true north, do not use, only updating 2 times a second
- float deviation

  deviation relative to true north; do we use this in our calculations? Nope
- float variance

  variance relative to true north; do we use this in our calculations? Nope
- float bspeed

  Boat's speed in km/h.
- float bspeedk

  Boat's speed in knots.
- float wind_angl

  wind angle, (relative to boat I believe, could be north, check this)
- float wind_velocity

  wind velocity in knots
- float headingc

  Heading from compass.
- float pitch

  pitch
- float roll

  roll
- float trueWind

  wind direction calculated at checkteck
- int rudderVal

  variables for transmiting data
- int jibVal

variables for transmiting data

- int mainVal

  variables for transmiting data
- float headingVal

  where we are going, temporary compass smoothing test
- float distanceVal

  distance to next waypoint one-shots, no averaging, present conditions
- float heading_newest

  heading relative to true north
- float wind_angl_newest

  wind angle, (relative to boat)
- SoftwareSerial servo_ser = SoftwareSerial(7, txPin)

  Pololu for connecting via a nonbuffered serial port to pololu -output only.
- int rudderDir = -1

  global for reversing rudder if we are parking lot testing
- int points

  max waypoints selected for travel
- int point

  point for sail to waypoint in menu
- int currentPoint = 0

  current waypoint on course of travel
- int StraightSailDirection

  Menu hack globals.
- int CurrentSelection

  Menu hack globals.
- long startTime

  station-keeping global
- int stationCounter

  station-keeping global
- boolean timesUp

  station-keeping global
- int StationKeepingTimeInBox = 270000
- boolean tacking
- int tackingSide
- int ironTime
- int errorCode

5.2.1 Detailed Description

Variables for Serial Communications.

Warning

When testing by sending strings through the serial monitor, you need to select "newline" ending from the dropdown beside the baud

5.2.2 Variable Documentation

5.2.2.1 float bspeed

Boat's speed in km/h.

Definition at line 152 of file sailcode_alpha6.pde.

5.2.2.2 float bspeedk

Boat's speed in knots.

Definition at line 153 of file sailcode_alpha6.pde.

5.2.2.3 int currentPoint = 0

current waypoint on course of travel

Definition at line 180 of file sailcode_alpha6.pde.

5.2.2.4 int CurrentSelection

Menu hack globals.

Definition at line 184 of file sailcode_alpha6.pde.

5.2.2.5 float deviation

deviation relative to true north; do we use this in our calculations? Nope

Definition at line 149 of file sailcode_alpha6.pde.

5.2.2.6 float distanceVal

distance to next waypoint one-shots, no averaging, present conditions

Definition at line 170 of file sailcode_alpha6.pde.

5.2.2.7 int errorCode

Definition at line 201 of file sailcode_alpha6.pde.

5.2.2.8 int extraCompassData = 0

Definition at line 135 of file sailcode_alpha6.pde.

5.2.2.9 char extraCompassDataArray[LONGEST_NMEA]

Definition at line 144 of file sailcode_alpha6.pde.

5.2.2.10 int extraWindData = 0

Contains extra data from the Wind Sensor.

Warning

'clear' the extra global data buffer, because any data wrapping around will be destroyed by clearing the buffer

Definition at line 134 of file sailcode_alpha6.pde.

5.2.2.11 char extraWindDataArray[LONGEST_NMEA]

a buffer to store roll-over data in case this data is fetched mid-line

Definition at line 143 of file sailcode_alpha6.pde.

5.2.2.12 float heading

heading relative to true north, do not use, only updating 2 times a second

Definition at line 148 of file sailcode_alpha6.pde.

5.2.2.13 float heading_newest

heading relative to true north

Definition at line 171 of file sailcode_alpha6.pde.

5.2.2.14 float headingc

Heading from compass.

Definition at line 160 of file sailcode_alpha6.pde.

5.2.2.15 float headingVal

where we are going, temporary compass smoothing test

Definition at line 169 of file sailcode_alpha6.pde.

5.2.2.16 int ironTime

Definition at line 198 of file sailcode_alpha6.pde.

5.2.2.17 int jibVal

variables for transmiting data

Definition at line 167 of file sailcode_alpha6.pde.

5.2.2.18 int mainVal

variables for transmiting data

Definition at line 168 of file sailcode_alpha6.pde.

5.2.2.19 float pitch

pitch

Definition at line 161 of file sailcode_alpha6.pde.

5.2.2.20 int point

point for sail to waypoint in menu

Definition at line 179 of file sailcode_alpha6.pde.

5.2.2.21 int points

max waypoints selected for travel

Definition at line 178 of file sailcode_alpha6.pde.

5.2.2.22 float roll

roll

Definition at line 162 of file sailcode_alpha6.pde.

5.2.2.23 int rudderDir = -1

global for reversing rudder if we are parking lot testing

Definition at line 177 of file sailcode_alpha6.pde.

5.2.2.24 int rudderVal

variables for transmiting data

Definition at line 166 of file sailcode_alpha6.pde.

5.2.2.25 int savedCompassChecksum = 0

Definition at line 140 of file sailcode_alpha6.pde.

5.2.2.26 int savedCompassXorState = 0

Definition at line 141 of file sailcode_alpha6.pde.

5.2.2.27 int savedWindChecksum = 0

clear the global saved XOR value

Definition at line 137 of file sailcode_alpha6.pde.

5.2.2.28 int savedWindXorState = 0

clear the global saved XORstate value

Definition at line 139 of file sailcode_alpha6.pde.

5.2.2.29 SoftwareSerial servo_ser = SoftwareSerial(7, txPin)

Pololu for connecting via a nonbuffered serial port to pololu -output only.

Definition at line 175 of file sailcode_alpha6.pde.

5.2.2.30 long startTime

station-keeping global

Definition at line 187 of file sailcode_alpha6.pde.

5.2.2.31 int stationCounter

station-keeping global

Definition at line 188 of file sailcode_alpha6.pde.

5.2.2.32 int StationKeepingTimeInBox = 270000

Definition at line 193 of file sailcode_alpha6.pde.

5.2.2.33 int StraightSailDirection

Menu hack globals.

Definition at line 183 of file sailcode_alpha6.pde.

5.2.2.34 boolean tacking

Definition at line 196 of file sailcode_alpha6.pde.

5.2.2.35 int tackingSide

Definition at line 197 of file sailcode_alpha6.pde.

5.2.2.36 boolean timesUp

station-keeping global

Definition at line 189 of file sailcode_alpha6.pde.

5.2.2.37 float trueWind

wind direction calculated at checkteck

Definition at line 163 of file sailcode_alpha6.pde.

5.2.2.38 float variance

variance relative to true north; do we use this in our calculations? Nope

Definition at line 150 of file sailcode_alpha6.pde.

5.2.2.39 float wind_angl

wind angle, (relative to boat I believe, could be north, check this)

Definition at line 156 of file sailcode_alpha6.pde.

5.2.2.40 float wind_angl_newest

wind angle, (relative to boat)

Definition at line 172 of file sailcode_alpha6.pde.

5.2.2.41 float wind_velocity

wind velocity in knots

Definition at line 157 of file sailcode_alpha6.pde.

Chapter 6

Data Structure Documentation

6.1 points Struct Reference

`#include <LocationStruct.h>`

Data Fields

- double latDeg
- double latMin
- double lonDeg
- double lonMin

6.1.1 Detailed Description

Definition at line 4 of file LocationStruct.h.

6.1.2 Field Documentation

6.1.2.1 double latDeg

Definition at line 5 of file LocationStruct.h.

6.1.2.2 double latMin

Definition at line 6 of file LocationStruct.h.

6.1.2.3 double lonDeg

Definition at line 7 of file LocationStruct.h.

6.1.2.4 double lonMin

Definition at line 8 of file LocationStruct.h.

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

- /Users/allgood38/Desktop/qmast/sailcode_alpha6/LocationStruct.h

Chapter 7

File Documentation

7.1 /Users/allgood38/Desktop/qmast/sailcode_alpha6/DataAcquisition.pde File Reference

Functions

- void sensorData (int bufferLength, char device)
- void setErrorBit (int aBit)
- void clearErrorBit (int aBit)
- int checkErrorBit (int aBit)

7.1.1 Function Documentation

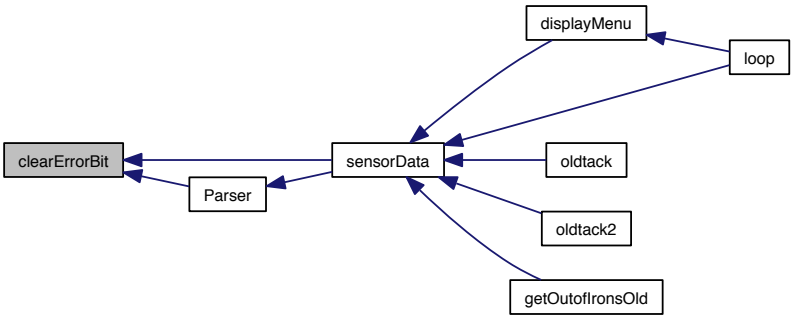
7.1.1.1 int checkErrorBit (int *aBit*)

Definition at line 213 of file DataAcquisition.pde.

7.1.1.2 void clearErrorBit (int *aBit*)

Definition at line 208 of file DataAcquisition.pde.

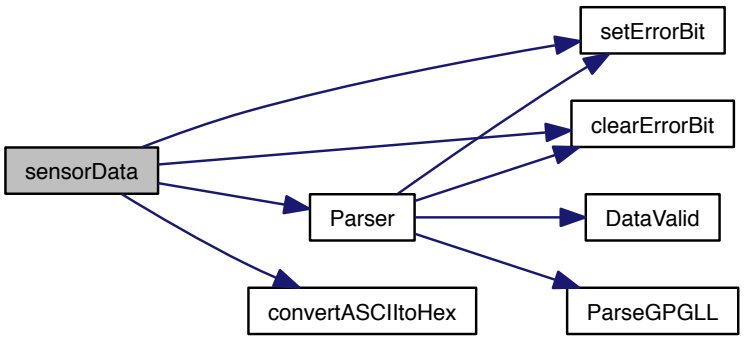
Here is the caller graph for this function:



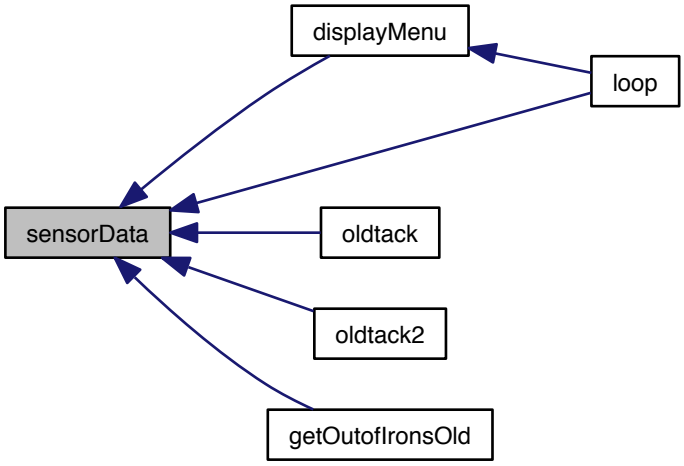
7.1.1.3 void sensorData (int *bufferLength*, char *device*)

Definition at line 4 of file DataAcquisition.pde.

Here is the call graph for this function:



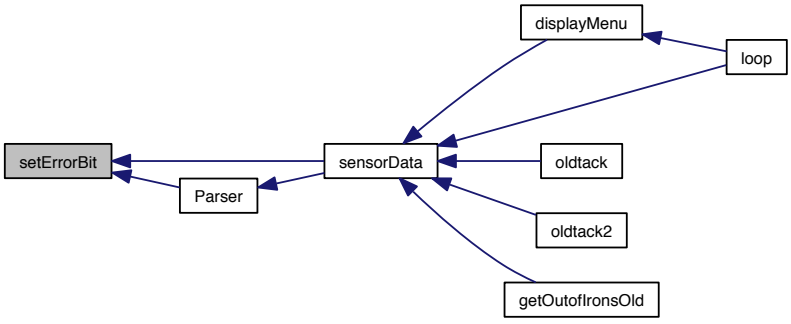
Here is the caller graph for this function:



7.1.1.4 void setErrorBit (int *aBit*)

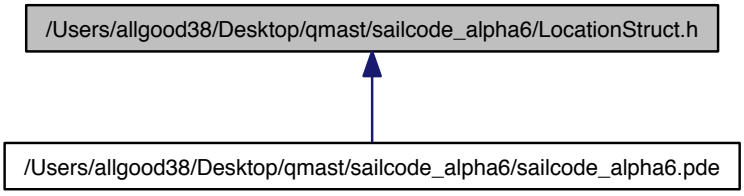
Definition at line 203 of file DataAcquisition.pde.

Here is the caller graph for this function:



7.2 /Users/allgood38/Desktop/qmast/sailcode_alpha6/LocationStruct.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct points

Variables

- points waypoints [10]
- points stationPoints [4]
- points floatingStationPoints [4]
- points coursePoints [10]
- points clearPoints
- points boatLocation
- points stayPoint

7.2.1 Variable Documentation

7.2.1.1 points boatLocation

Definition at line 35 of file LocationStruct.h.

7.2.1.2 points clearPoints

Initial value:

```
{
 0,0,0,0}
```

Definition at line 32 of file LocationStruct.h.

7.2.1.3 points coursePoints[10]

Initial value:

```
{
{0,0,0,0},
{0,0,0,0},
{0,0,0,0},
{0,0,0,0},
{0,0,0,0},
{0,0,0,0},
{0,0,0,0},
{0,0,0,0},
{0,0,0,0}}
```

Definition at line 21 of file LocationStruct.h.

7.2.1.4 points floatingStationPoints[4]

Definition at line 20 of file LocationStruct.h.

7.2.1.5 points stationPoints[4]

Definition at line 18 of file LocationStruct.h.

7.2.1.6 points stayPoint

Definition at line 37 of file LocationStruct.h.

7.2.1.7 points waypoints[10]

Initial value:

```
{
 {44.0,13.6927,-76.0,-29.5175},
 {38.0,58.9443,-76.0,-28.7383},
 {38.0,58.9515,-76.0,-28.7127}}
```

Definition at line 12 of file LocationStruct.h.

7.3 /Users/allgood38/Desktop/qmast/sailcode_alpha6/Menu.pde - File Reference

Functions

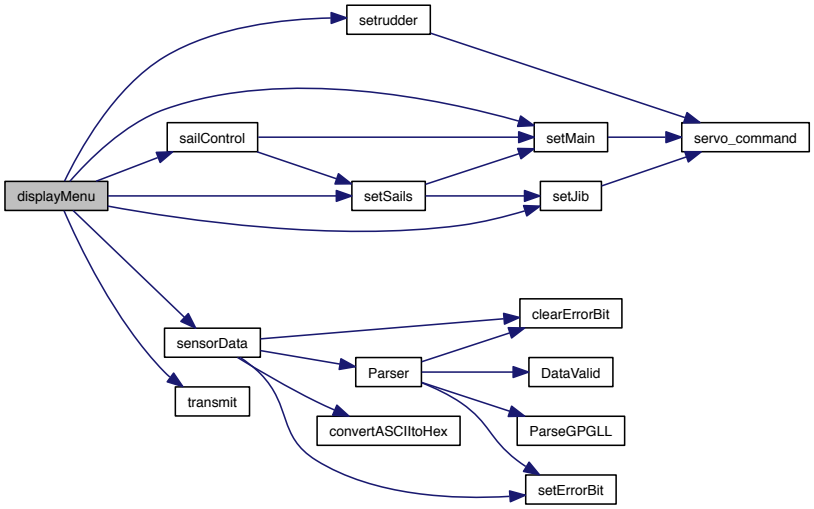
- int displayMenu ()

7.3.1 Function Documentation

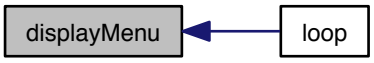
7.3.1.1 int displayMenu ( )

Definition at line 1 of file Menu.pde.

Here is the call graph for this function:



Here is the caller graph for this function:



7.4 /Users/allgood38/Desktop/qmast/sailcode_alpha6/MotorControl-Functions.pde File Reference

Functions

- void servo_command (int whichservo, int position, byte longRange)

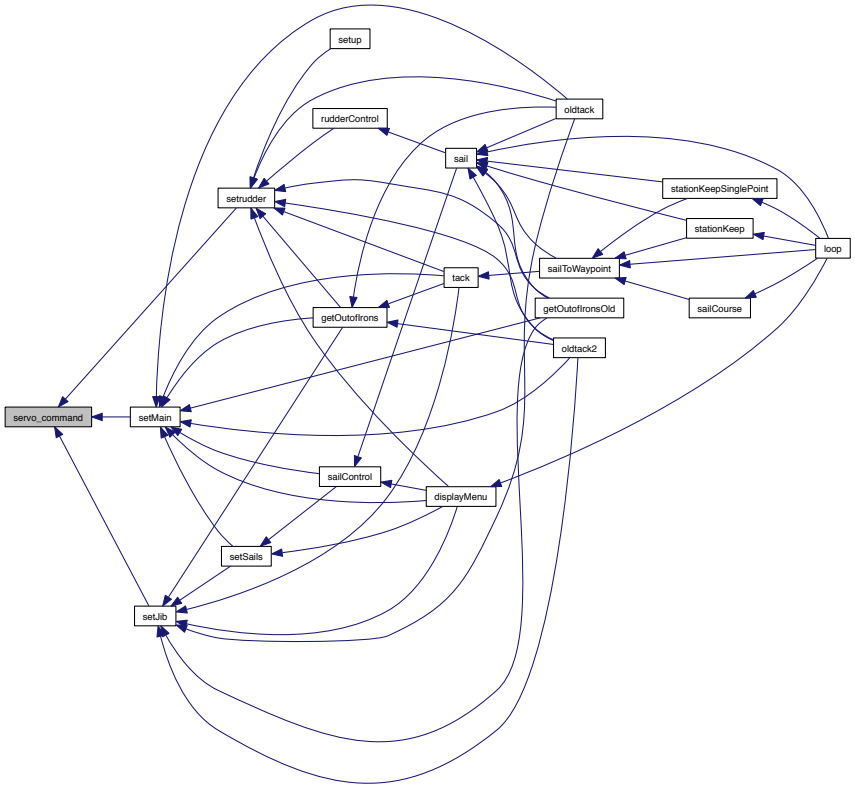
- void setrudder (float ang)
- void setSails (float ang)
- void setJib (float ang)
- void setMain (float ang)

7.4.1 Function Documentation

7.4.1.1 void servo_command (int *whichservo,* int *position,* byte *longRange*)

Definition at line 6 of file MotorControlFunctions.pde.

Here is the caller graph for this function:

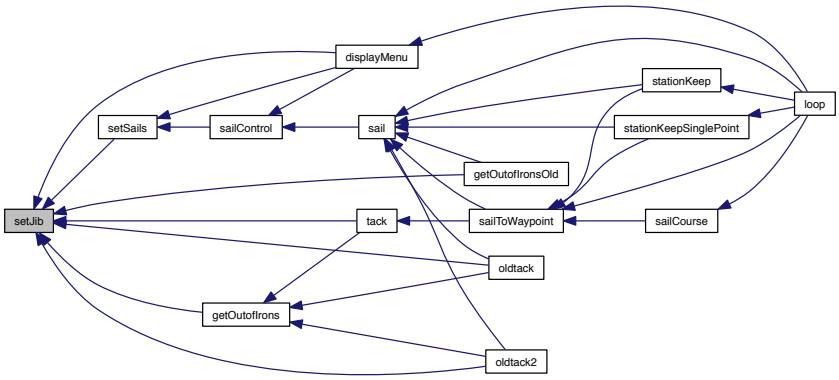


7.4.1.2 void setJib (float *ang*)

Definition at line 37 of file MotorControlFunctions.pde.

Here is the call graph for this function:

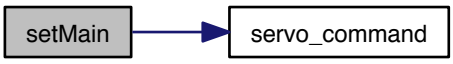
Here is the caller graph for this function:



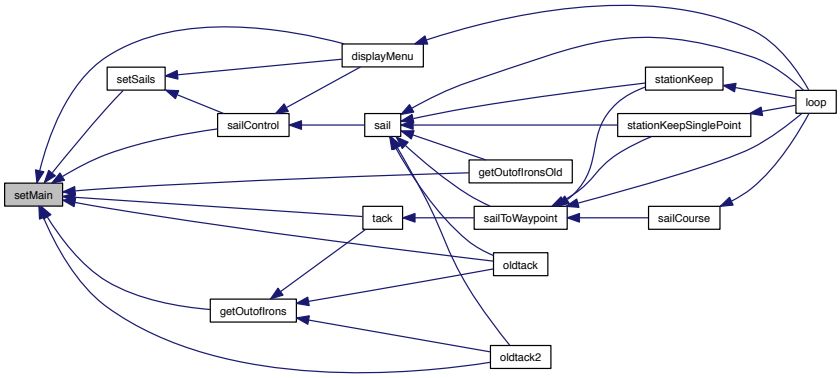
7.4.1.3 void setMain (float *ang*)

Definition at line 49 of file MotorControlFunctions.pde.

Here is the call graph for this function:



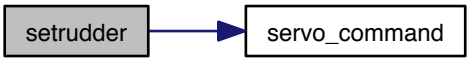
Here is the caller graph for this function:



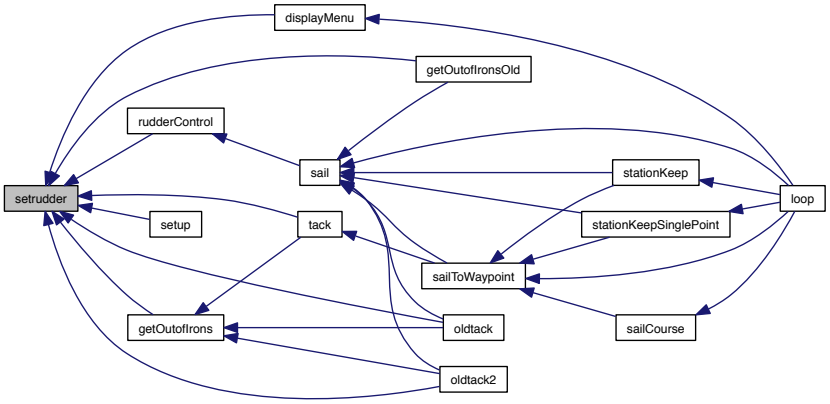
7.4.1.4 void setrudder (float *ang*)

Definition at line 17 of file MotorControlFunctions.pde.

Here is the call graph for this function:



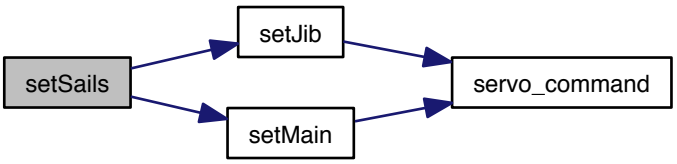
Here is the caller graph for this function:



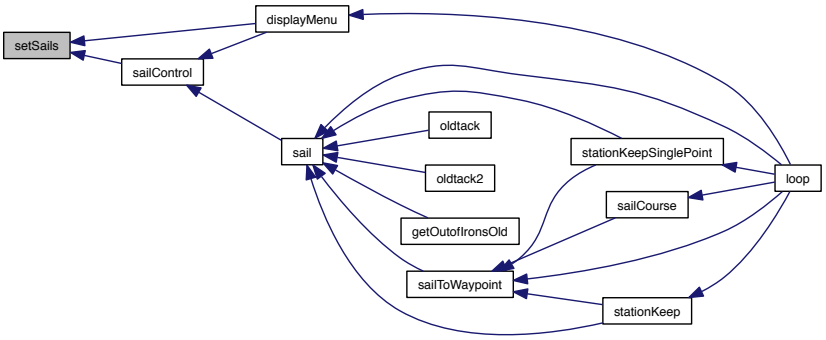
7.4.1.5 void setSails (float *ang*)

Definition at line 30 of file MotorControlFunctions.pde.

Here is the call graph for this function:



Here is the caller graph for this function:



7.5 /Users/allgood38/Desktop/qmast/sailcode_alpha6/NavigationCode.pde File Reference

Functions

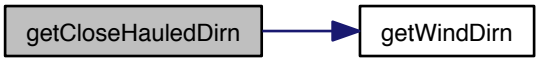
- double GPSdistance (struct points location1, struct points location2)
- int getWaypointDirn (struct points waypoint)
- int getCloseHauledDirn ()
- int getOppositeCloseHauledDirn ()
- int getWindDirn ()

7.5.1 Function Documentation

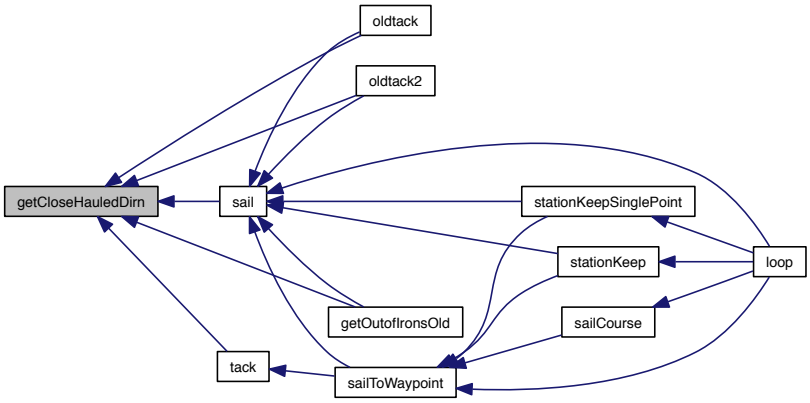
7.5.1.1 int getCloseHauledDirn ( )

Definition at line 42 of file NavigationCode.pde.

Here is the call graph for this function:



Here is the caller graph for this function:



7.5.1.2 int getOppositeCloseHauledDirn ( )

Definition at line 60 of file NavigationCode.pde.

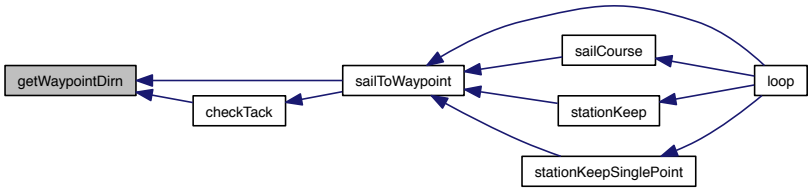
Here is the call graph for this function:

7.5.1.3 int getWaypointDirn (struct points *waypoint*)

Definition at line 16 of file NavigationCode.pde.

Here is the call graph for this function:

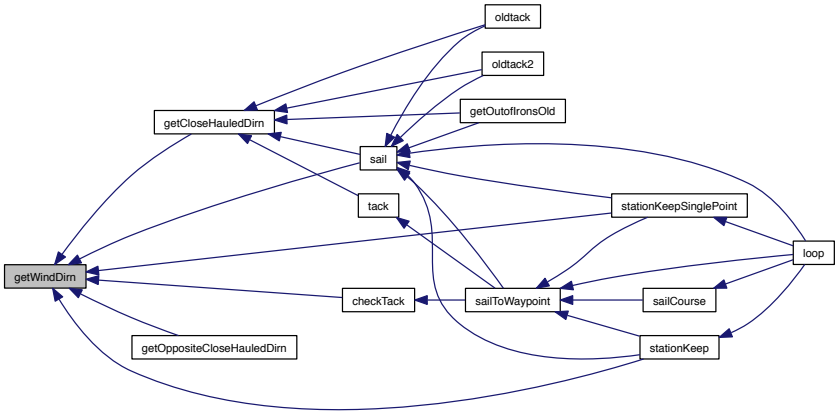
Here is the caller graph for this function:



7.5.1.4 int getWindDirn ( )

Definition at line 83 of file NavigationCode.pde.

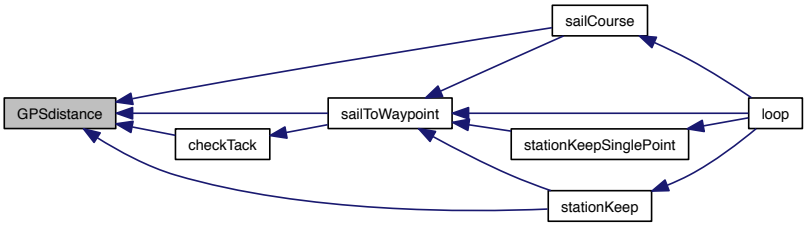
Here is the caller graph for this function:



7.5.1.5 double GPSdistance (struct points *location1*, struct points *location2*)

Definition at line 1 of file NavigationCode.pde.

Here is the caller graph for this function:



7.6 /Users/allgood38/Desktop/qmast/sailcode_alpha6/oldtack.pde - File Reference

Functions

- void oldtack ()
- void oldtack2 ()

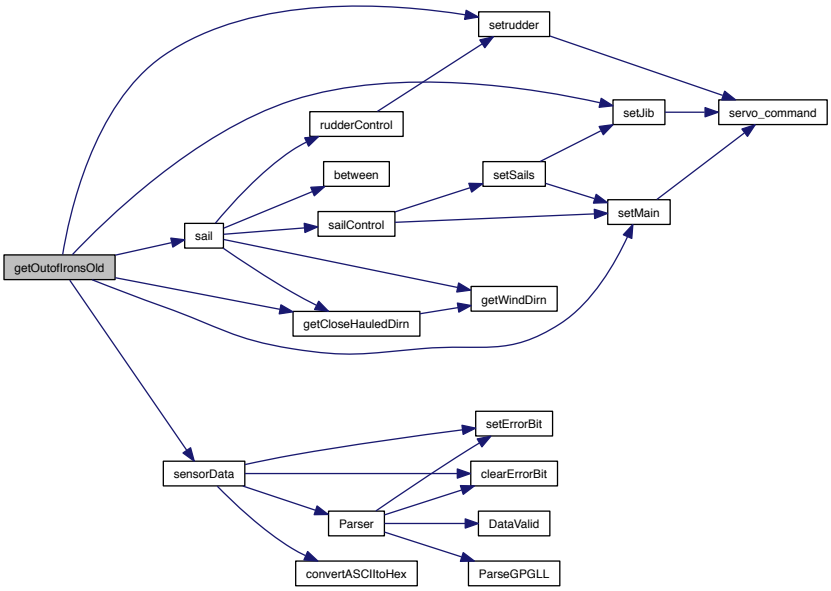
  newer revision
- void getOutofIronsOld (int tackside)

7.6.1 Function Documentation

7.6.1.1 void getOutofIronsOld (int *tackside*)

Definition at line 171 of file oldtack.pde.

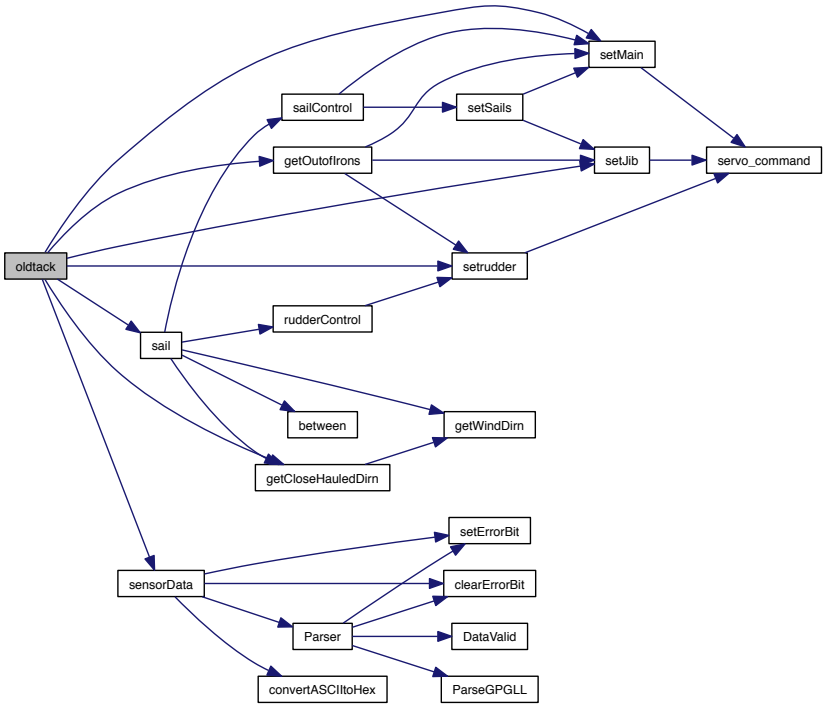
Here is the call graph for this function:



7.6.1.2 void oldtack ( )

Definition at line 6 of file oldtack.pde.

Here is the call graph for this function:

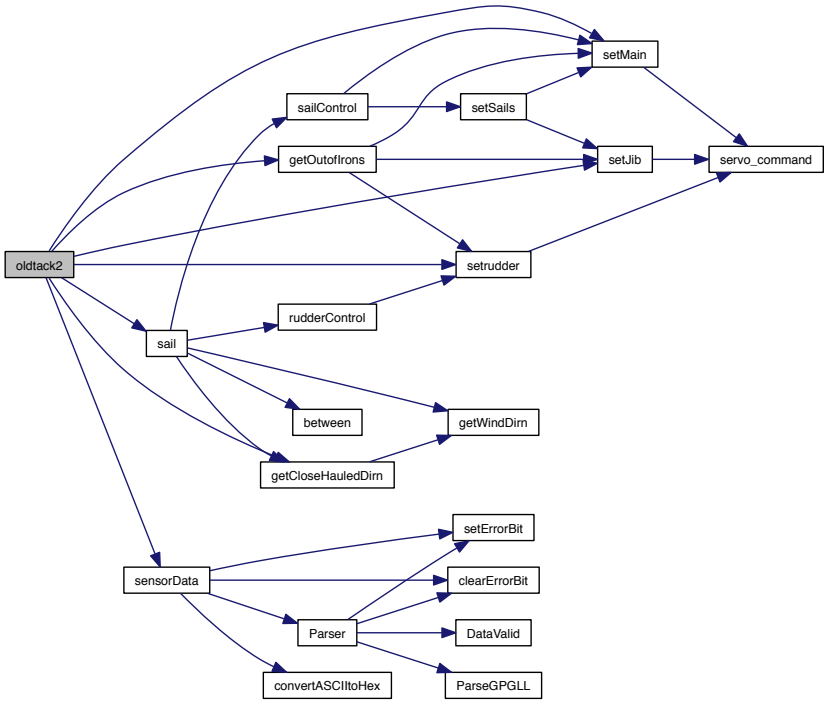


7.6.1.3 void oldtack2 ( )

newer revision

Definition at line 83 of file oldtack.pde.

Here is the call graph for this function:



7.7 /Users/allgood38/Desktop/qmast/sailcode_alpha6/ParsingFunctions.pde File Reference

Functions

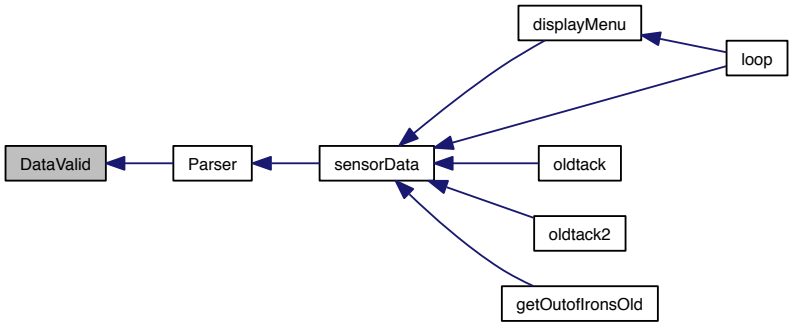
- int DataValid (char *val)
- void ParseGPGLL (char *GPGLL_string, double *degree, double *minute)
- int Parser (char *val)

7.7.1 Function Documentation

7.7.1.1 int DataValid (char * *val*)

Definition at line 5 of file ParsingFunctions.pde.

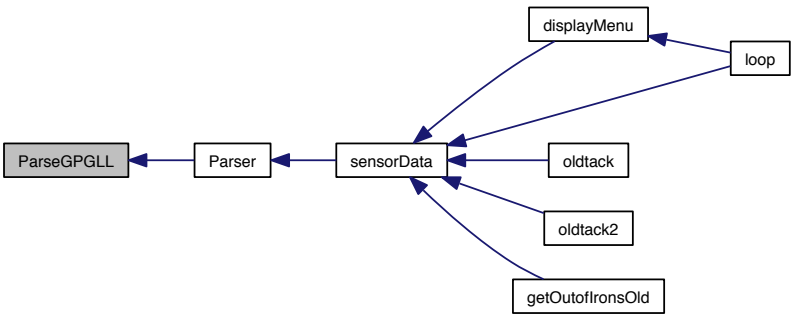
Here is the caller graph for this function:



7.7.1.2 void ParseGPGLL (char * *GPGLL_string,* double * *degree,* double * *minute*)

Definition at line 14 of file ParsingFunctions.pde.

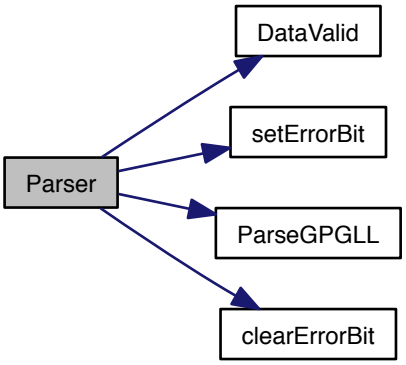
Here is the caller graph for this function:



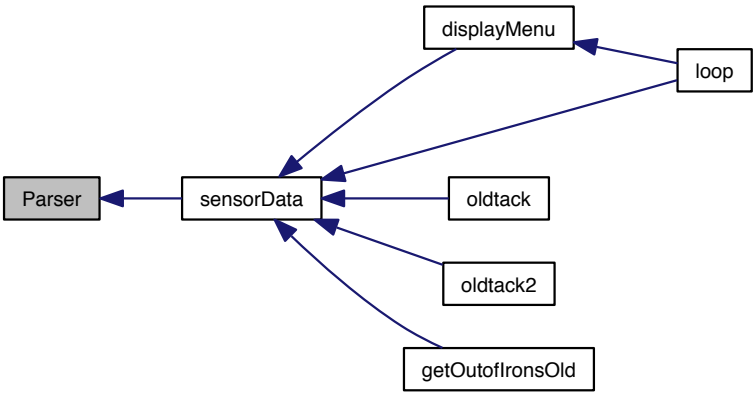
7.7.1.3 int Parser (char * *val*)

Definition at line 64 of file ParsingFunctions.pde.

Here is the call graph for this function:



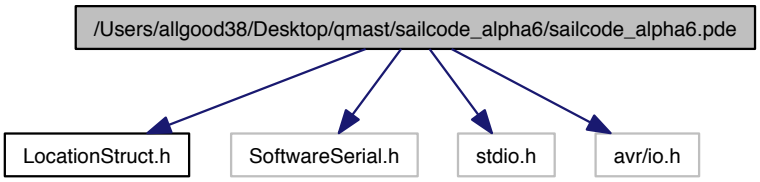
Here is the caller graph for this function:



7.8 /Users/allgood38/Desktop/qmast/sailcode_alpha6/sailcode_alpha6.pde File Reference

```
#include "LocationStruct.h" #include <SoftwareSerial.h> ×
```

`#include <stdio.h>` `#include <avr/io.h>` Include dependency graph for sailcode_alpha6.pde:



Defines

- #define TACKING_ANGLE 40

  Boat parameter constants in globalconstants.
- #define MARK_DISTANCE 4

  Course Navigation constants.
- #define STATION_KEEPING_RADIUS 15

  Station keeping navigation constants.
- #define WIND_CHANGE_THRESHOLD 10
- #define BUFF_MAX 511

  serial data constants
- #define DEGREE_TO_MINUTE 60

  Calculation constantes.
- #define LATITUDE_TO_METER 1855

  There are (approximately) 1855 meters in a minute of latitude everywhere; This isn't true for longitude, as it depends on the latitude.
- #define LONGITUDE_TO_METER 1314
- #define noDataBit 0

  no data, error error bit
- #define oldDataBit 1

  there is data, but buffer is full, error bit
- #define checksumBadBit 2

  indicates checksum fail on data
- #define twoCommasBit 3

  indicates that there were two commas in the data, and it has been discarded and not parsed
- #define rolloverDataBit 4

  Indicates data rolled over, not fast enough.
- #define badCompassDataBit 5

indicates that strtok did not return PTNTHTM, so we probably got bad data

- #define tooMuchRollBit 6

  indicates the boat is falling over
- #define badWindData 7

  indicates an error from the wind sensor
- #define badGpsData 8

  indicates error in gps data
- #define ALL_IN 0

  Constant which defines when the boat is "All In".
- #define ALL_OUT 100

  Constant which defines when the boat is "All Out".
- #define resetPin 8

  Pololu reset (digital pin on arduino)
- #define txPin 9

  Pololu serial pin (with SoftwareSerial library)
- #define SHORTEST_NMEA 5

  The shortest possible NMEA String.
- #define LONGEST_NMEA 120

  The longest possible NMEA String.

Functions

- void setup ()

  Standard Setup function for Arduino, set pins and create object instances.
- void loop ()

  Main Function, handles menu input and calls the core functions.

Variables

- int extraWindData = 0

  Contains extra data from the Wind Sensor.
- int extraCompassData = 0
- int savedWindChecksum = 0

  clear the global saved XOR value
- int savedWindXorState = 0

  clear the global saved XORstate value
- int savedCompassChecksum = 0
- int savedCompassXorState = 0
- char extraWindDataArray [LONGEST_NMEA]

  a buffer to store roll-over data in case this data is fetched mid-line
- char extraCompassDataArray [LONGEST_NMEA]
- float heading

  heading relative to true north, do not use, only updating 2 times a second

- float deviation

  deviation relative to true north; do we use this in our calculations? Nope
- float variance

  variance relative to true north; do we use this in our calculations? Nope
- float bspeed

  Boat's speed in km/h.
- float bspeedk

  Boat's speed in knots.
- float wind_angl

  wind angle, (relative to boat I believe, could be north, check this)
- float wind_velocity

  wind velocity in knots
- float headingc

  Heading from compass.
- float pitch

  pitch
- float roll

  roll
- float trueWind

  wind direction calculated at checkteck
- int rudderVal

  variables for transmiting data
- int jibVal

  variables for transmiting data
- int mainVal

  variables for transmiting data
- float headingVal

  where we are going, temporary compass smoothing test
- float distanceVal

  distance to next waypoint one-shots, no averaging, present conditions
- float heading_newest

  heading relative to true north
- float wind_angl_newest

  wind angle, (relative to boat)
- SoftwareSerial servo_ser = SoftwareSerial(7, txPin)

  Pololu for connecting via a nonbuffered serial port to pololu -output only.
- int rudderDir = -1

  global for reversing rudder if we are parking lot testing
- int points

  max waypoints selected for travel
- int point

  point for sail to waypoint in menu
- int currentPoint = 0

current waypoint on course of travel

- int StraightSailDirection

  Menu hack globals.

- int CurrentSelection

  Menu hack globals.

- long startTime

  station-keeping global

- int stationCounter

  station-keeping global

- boolean timesUp

  station-keeping global

- int StationKeepingTimeInBox = 270000
- boolean tacking
- int tackingSide
- int ironTime
- int errorCode

7.8.1 Function Documentation

7.8.1.1 void loop ( )

Main Function, handles menu input and calls the core functions.

A lot of documentation should probably be written for this, but all I know right now is that it contains the switch statement in order to call the functions. if menu returned 0, any updating happened in the menu function itself and we want the code to just keep doing what it was doing before (e.g. setting RC mode)

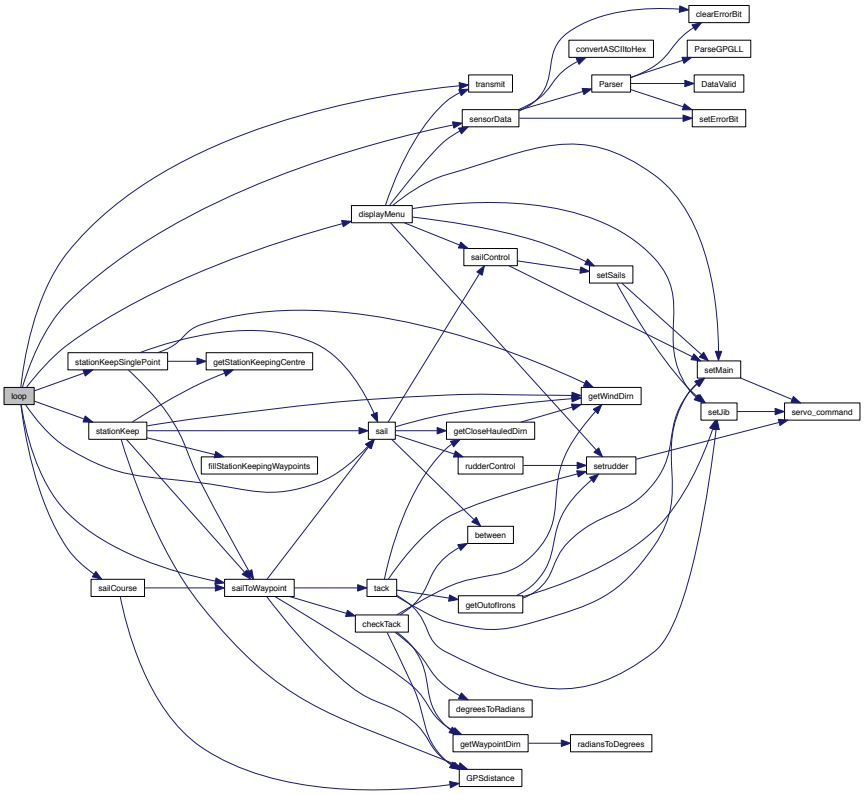
Straight Sail towards N,S,E,W as 0, 180, 90, 270. No sail control.

Straightsail can no longer be called in isolation, needs sailtoWaypoint which keeps track of when tacking is necessary

stationskeeps around a single spot in the middle of the square

Definition at line 294 of file sailcode_alpha6.pde.

Here is the call graph for this function:



7.8.1.2 void setup ( )

Standard Setup function for Arduino, set pins and create object instances.

The setup function is the first function to be called. When it exits, the loop function is immediately called, where the program remains.

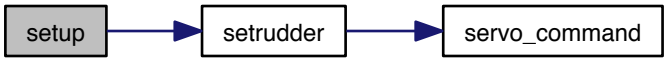
Warning

> In order for Pololu to work, it needs to be manually reset within the code. Because of this, a 2 second delay is present to allow the board to start-up before the reset occurs.

Change wind to send 5 times a second default for now, need to make sure we can get everything out of the buffer

Definition at line 214 of file sailcode_alpha6.pde.

Here is the call graph for this function:



7.9 /Users/allgood38/Desktop/qmast/sailcode_alpha6/SailingLogic.pde File Reference

Functions

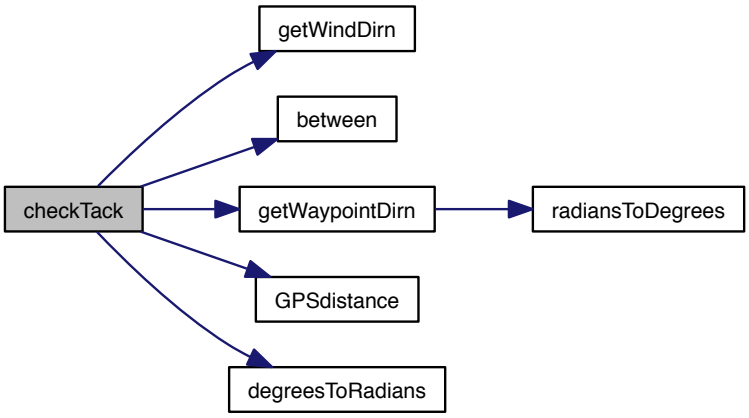
- void sailCourse ()
- void sailToWaypoint (struct points waypoint)
- void sail (int waypointDirn)
- boolean checkTack (int corridorHalfWidth, struct points waypoint)
- void sailControl ()
- int rudderControl (int directionError)

7.9.1 Function Documentation

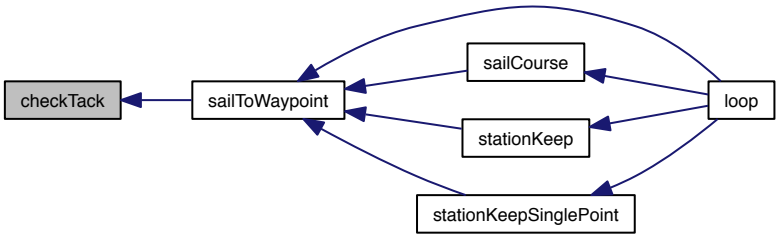
7.9.1.1 boolean checkTack (int *corridorHalfWidth,* struct points *waypoint*)

Definition at line 63 of file SailingLogic.pde.

Here is the call graph for this function:



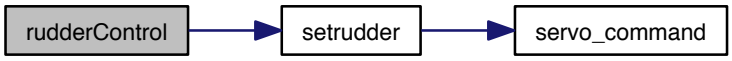
Here is the caller graph for this function:



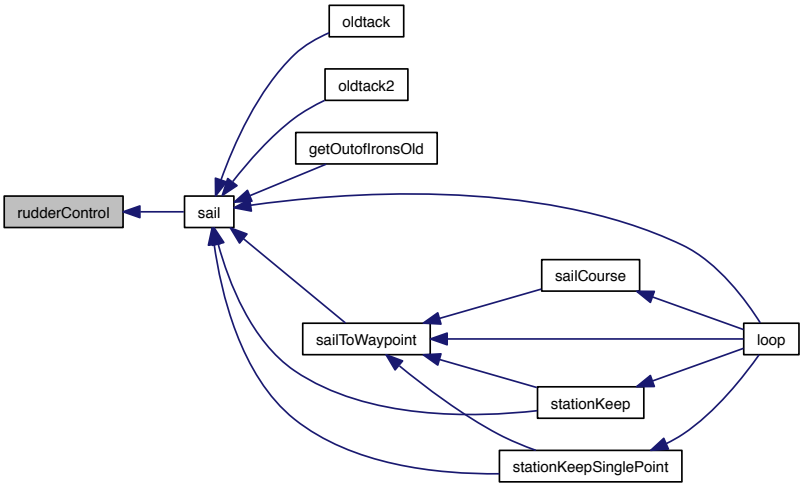
7.9.1.2 int rudderControl (int *directionError*)

Definition at line 117 of file SailingLogic.pde.

Here is the call graph for this function:



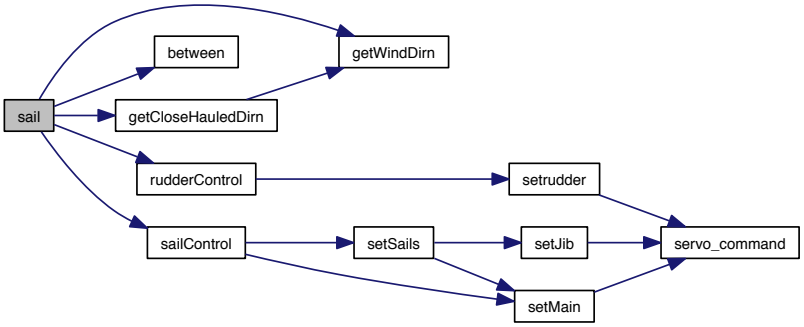
Here is the caller graph for this function:



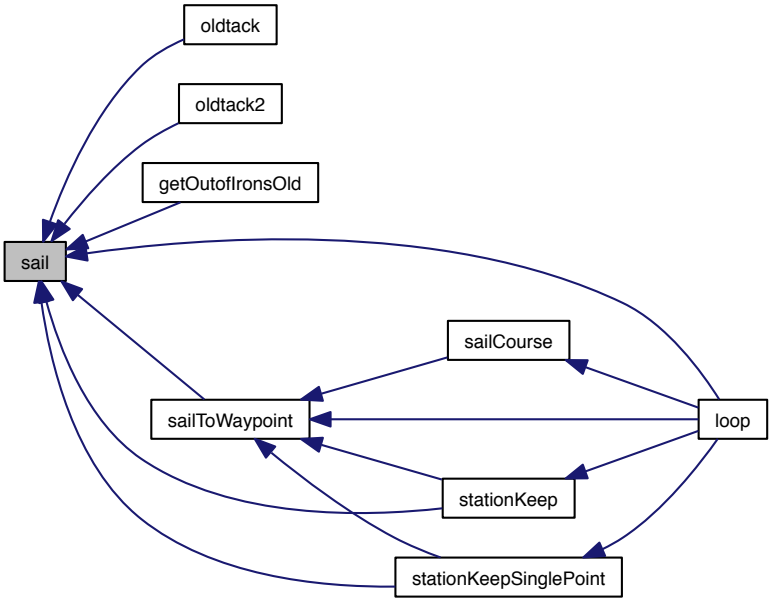
7.9.1.3 void sail (int *waypointDirn*)

Definition at line 40 of file SailingLogic.pde.

Here is the call graph for this function:



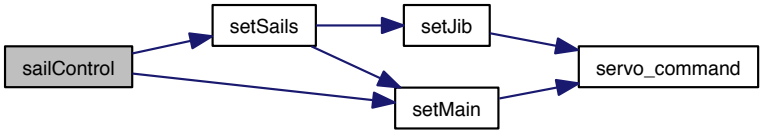
Here is the caller graph for this function:



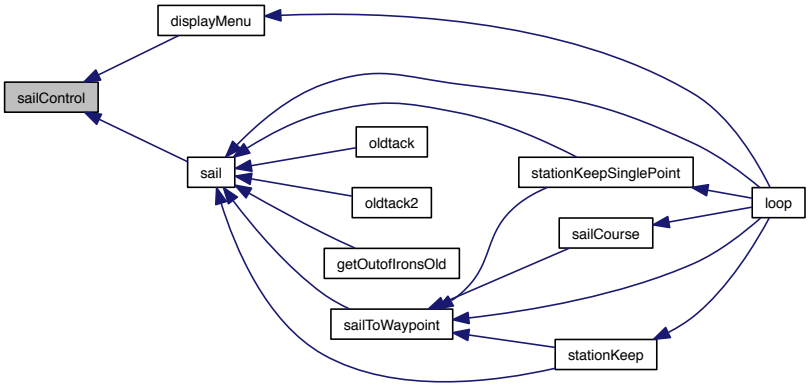
7.9.1.4 void sailControl ( )

Definition at line 99 of file SailingLogic.pde.

Here is the call graph for this function:



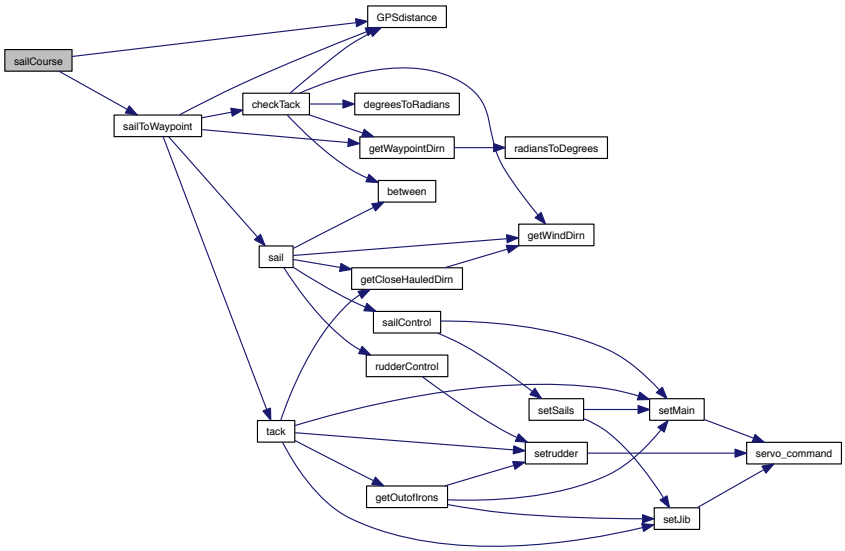
Here is the caller graph for this function:



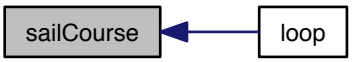
7.9.1.5 void sailCourse ( )

Definition at line 4 of file SailingLogic.pde.

Here is the call graph for this function:



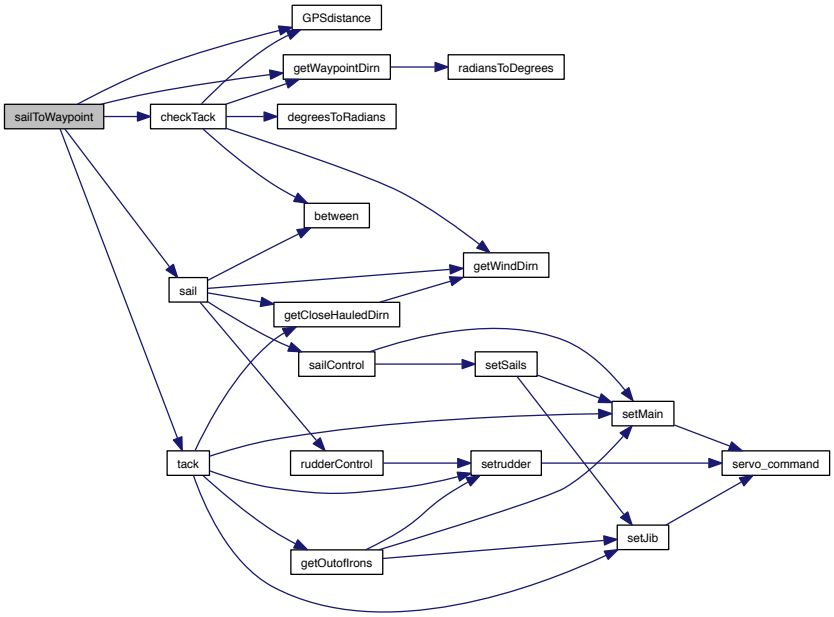
Here is the caller graph for this function:



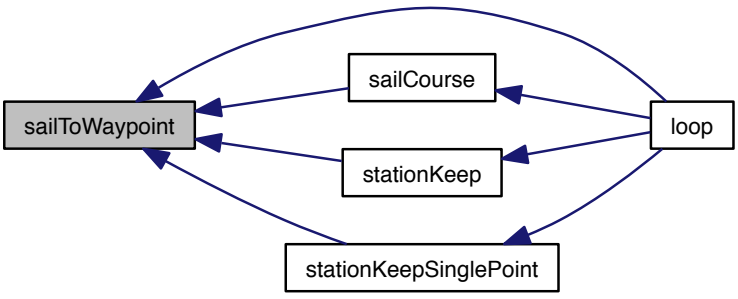
7.9.1.6 void sailToWaypoint (struct **points** *waypoint*)

Definition at line 21 of file SailingLogic.pde.

Here is the call graph for this function:



Here is the caller graph for this function:



7.10 /Users/allgood38/Desktop/qmast/sailcode_alpha6/StationKeeping.pde File Reference

Functions

- void getStationKeepingCentre (double *centreLatMin, double *centreLonMin)
- void fillStationKeepingWaypoints (double centreLatMin, double centreLonMin, int windBearing)
- int stationKeep ()
- void stationKeepSinglePoint ()

7.10.1 Function Documentation

7.10.1.1 void fillStationKeepingWaypoints (double *centreLatMin*, double *centreLonMin*, int *windBearing*)

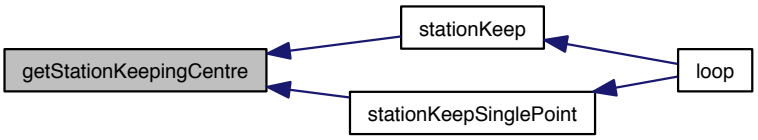
Definition at line 22 of file StationKeeping.pde.

Here is the caller graph for this function:

7.10.1.2 void getStationKeepingCentre (double * *centreLatMin*, double * *centreLonMin*)

Definition at line 2 of file StationKeeping.pde.

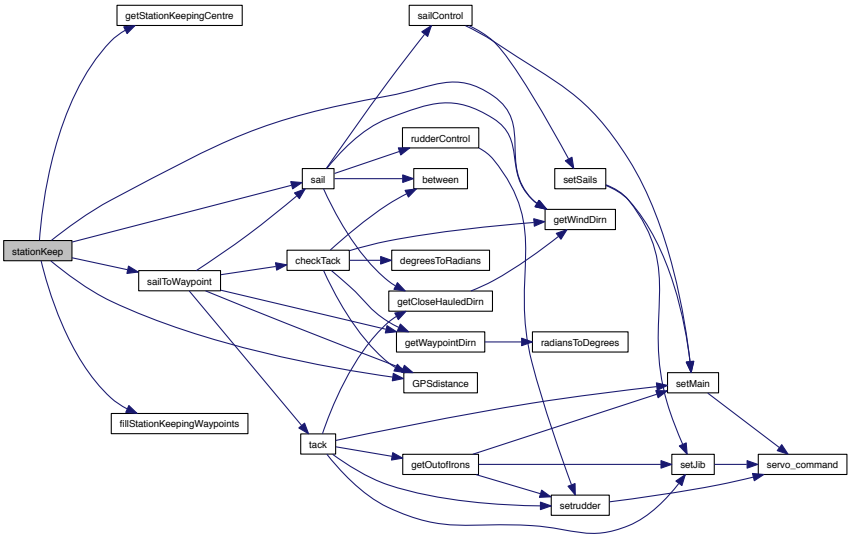
Here is the caller graph for this function:



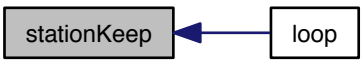
7.10.1.3 int stationKeep ( )

Definition at line 43 of file StationKeeping.pde.

Here is the call graph for this function:



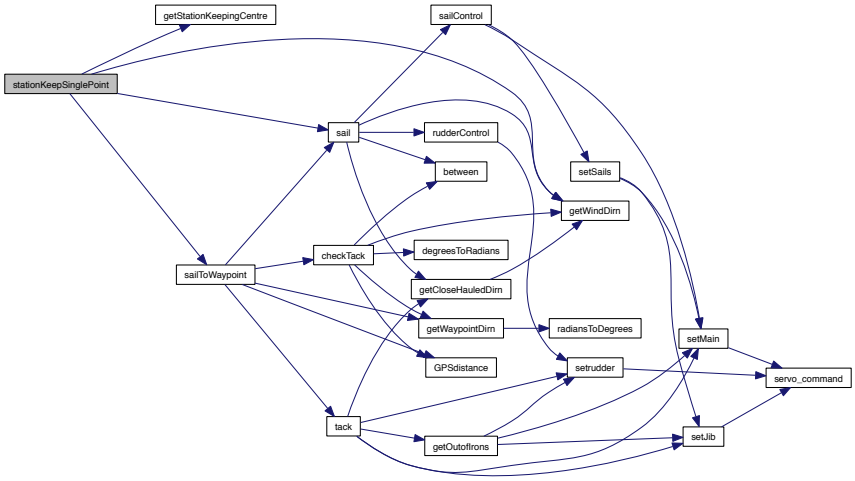
Here is the caller graph for this function:



7.10.1.4 void stationKeepSinglePoint ( )

Definition at line 106 of file StationKeeping.pde.

Here is the call graph for this function:



Here is the caller graph for this function:

7.11 /Users/allgood38/Desktop/qmast/sailcode_alpha6/Tack.pde - File Reference

Functions

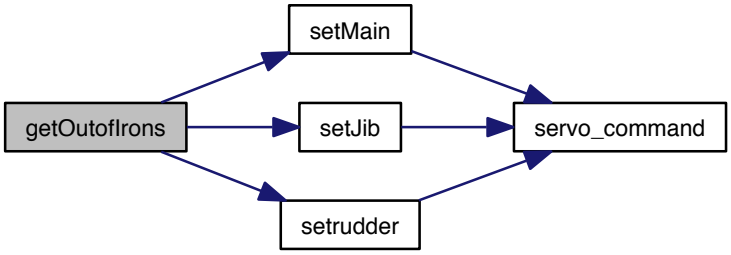
- void tack ()
- void getOutofIrons (int tackside)

7.11.1 Function Documentation

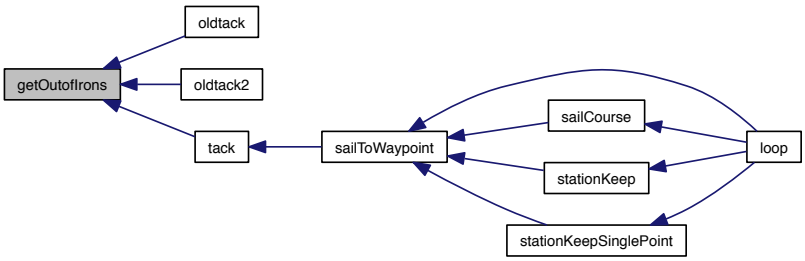
7.11.1.1 void getOutofIrons (int *tackside*)

Definition at line 61 of file Tack.pde.

Here is the call graph for this function:



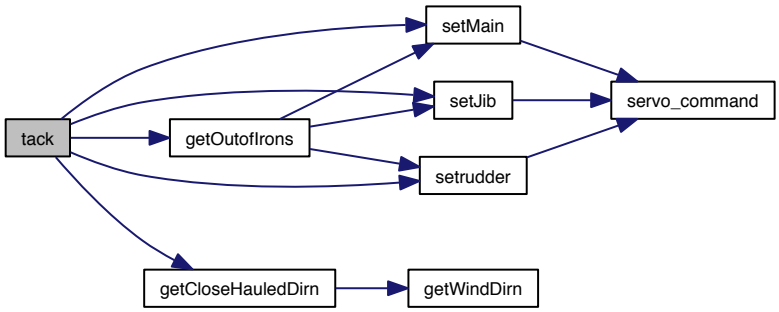
Here is the caller graph for this function:



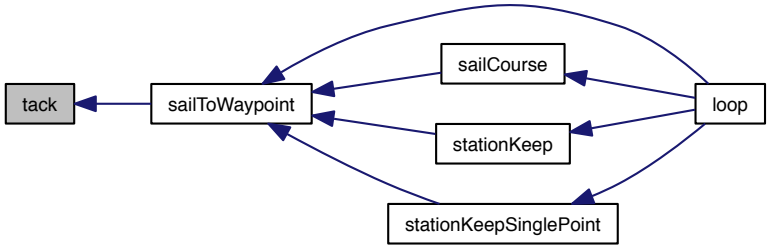
7.11.1.2 void tack ( )

Definition at line 7 of file Tack.pde.

Here is the call graph for this function:



Here is the caller graph for this function:



7.12 /Users/allgood38/Desktop/qmast/sailcode_alpha6/Testing_-Functions.pde File Reference

7.13 /Users/allgood38/Desktop/qmast/sailcode_alpha6/Transmit.pde File Reference

Functions

- void transmit (void)

  The transmit function is used to communicate with LabView? through a series of preset strings, which are represented by the graphical gauages?

- void relayData ()

7.13.1 Function Documentation

7.13.1.1 void relayData ( )

Similar to the transmit function except the output contains information which is not read by LabView, so codes are not needed?

Latitude and longitude of boat's location, split into more precise degrees and minutes, to fit into a float

Definition at line 56 of file Transmit.pde.

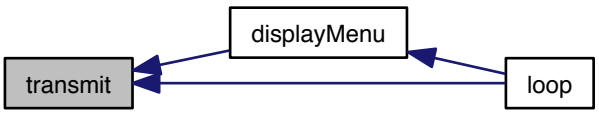
7.13.1.2 void transmit (void)

The transmit function is used to communicate with LabView? through a series of preset strings, which are represented by the graphical gauages?

Prints directly to the serial and takes input from global values

Definition at line 8 of file Transmit.pde.

Here is the caller graph for this function:



7.14 /Users/allgood38/Desktop/qmast/sailcode_alpha6/Utilities.pde File Reference

Functions

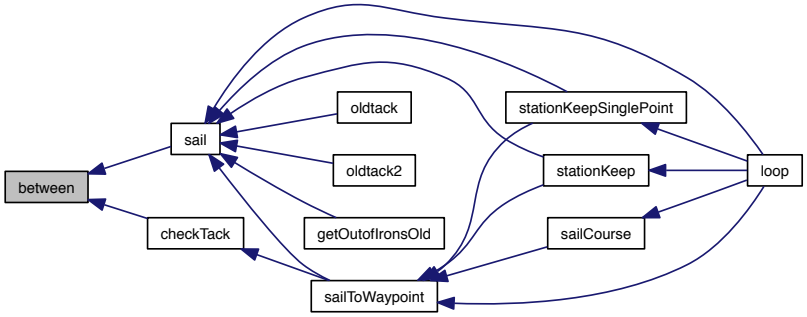
- float degreesToRadians (int angle)
- int radiansToDegrees (float angle)
- boolean between (int angle, int a, int b)
- char convertASCIItoHex (const char ch)

7.14.1 Function Documentation

7.14.1.1 boolean between (int *angle*, int *a*, int *b*)

Definition at line 12 of file Utilities.pde.

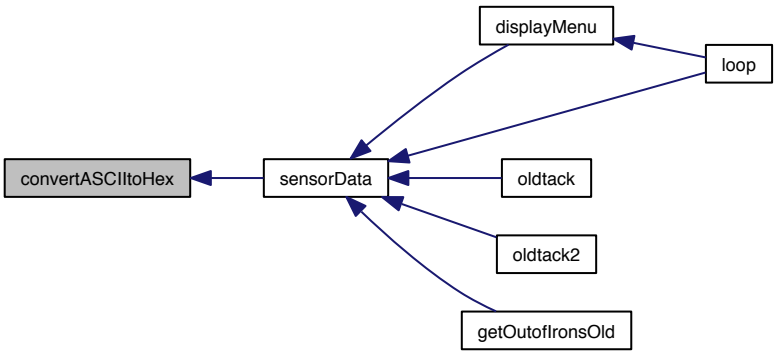
Here is the caller graph for this function:



7.14.1.2 char convertASCIItoHex (const char *ch*)

Definition at line 49 of file Utilities.pde.

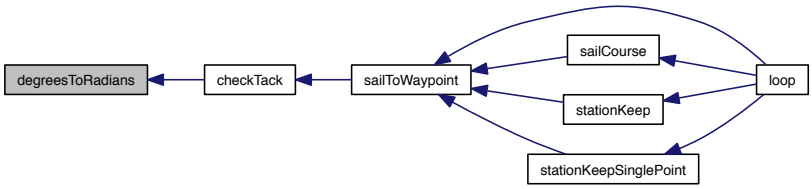
Here is the caller graph for this function:



7.14.1.3 float degreesToRadians (int *angle*)

Definition at line 3 of file Utilities.pde.

Here is the caller graph for this function:



7.14.1.4 int radiansToDegrees (float *angle*)

Definition at line 8 of file Utilities.pde.

Here is the caller graph for this function: