ugv_course_libs

Generated by Doxygen 1.8.6

Sun Dec 13 2015 20:36:24

Class Documentation

1 LatLon Class Reference

Class to contain a lat/lon point.

`#include <gps_conv.h>`

Public Member Functions

- LatLon ()

  Empty constructor initializes coordinates to zero.
- LatLon (const tf::Vector3 &ecef)

  Initialize coordinates to equivalent ECEF point.
- LatLon (double lat, double lon, double alt)

  Directly initialize the coordinates with lat, lon, and alt.
- LatLon (const sensor_msgs::NavSatFix &fix)

  Initialize coordinates using the data in the ROS message 'sensor_msgs::NavSatFix'.
- double getLat ()

  Returns the current latitude.
- double getLon ()

  Returns the current longitude.
- double getAlt ()

  Returns the current altitude.
- tf::Vector3 ecef ()

  Converts the current coordinates into ECEF.
- tf::Vector3 toEnu (const tf::Vector3 &ecef)

  Uses the current coordinates as a reference point, then converts an ECEF point into ENU coordinates centered at the reference.
- tf::Transform getTransform ()

  Computes and returns a transform from ECEF to an ENU frame centered at the current coordinates.
- void setEcef (const tf::Vector3 &ecef)

  Sets the current coordinates by converting ECEF to lat/lon.

1.1 Detailed Description

Class to contain a lat/lon point.

This class represents a geographic point in lat/lon. It also supports conversion to ENU, ECEF, and UTM Cartesian coordinate systems.

1.2 Constructor & Destructor Documentation

1.2.1 LatLon::LatLon (const tf::Vector3 & *ecef*) [inline]

Initialize coordinates to equivalent ECEF point.

Parameters

in	*ecef*	ECEF coordinates contained in a tf::Vector3

1.2.2 LatLon::LatLon (double *lat,* double *lon,* double *alt*) [inline]

Directly initialize the coordinates with lat, lon, and alt.

Parameters

in	*lat*	Latitude in degrees
in	*lon*	Longitude in degrees
in	*alt*	Altitude in meters

1.2.3 LatLon::LatLon (const sensor_msgs::NavSatFix & *fix*) [inline]

Initialize coordinates using the data in the ROS message 'sensor_msgs::NavSatFix'.

Parameters

in	*fix*	

1.3 Member Function Documentation

1.3.1 tf::Vector3 LatLon::ecef ( ) [inline]

Converts the current coordinates into ECEF.

Returns

ECEF coordinates as a tf::Vector3

1.3.2 double LatLon::getAlt ( ) [inline]

Returns the current altitude.

Returns

Altitude in meters

1.3.3 double LatLon::getLat ( ) [inline]

Returns the current latitude.

Returns

Latitude in degrees

1.3.4 double LatLon::getLon ( ) [inline]

Returns the current longitude.

Returns

Longitude in degrees

1.3.5 tf::Transform LatLon::getTransform ( ) [inline]

Computes and returns a transform from ECEF to an ENU frame centered at the current coordinates.

Returns

Translation and rotation represented in a tf::Transform

1.3.6 void LatLon::setEcef (const tf::Vector3 & *ecef*) [inline]

Sets the current coordinates by converting ECEF to lat/lon.

Parameters

in	*ecef*	ECEF point expressed as a tf::Vector3.

1.3.7 tf::Vector3 LatLon::toEnu (const tf::Vector3 & *ecef*) [inline]

Uses the current coordinates as a reference point, then converts an ECEF point into ENU coordinates centered at the reference.

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

- include/ugv_course_libs/gps_conv.h

2 UTMCoords Class Reference

Class to contain a UTM point.

`#include <gps_conv.h>`

Public Member Functions

- UTMCoords ()
  Empty constructor initializes everything to zero.
- UTMCoords (double x, double y, double z, int zone, int hemi)
  Initialize UTM coordinates directly.
- UTMCoords (const sensor_msgs::NavSatFix &fix)
  Initialize coordinates using the data in the ROS message 'sensor_msgs::NavSatFix'.
- UTMCoords (LatLon fix)
  Initialize coordinates using an instance of the LatLon class.
- double getX ()
  Returns the current UTM Easting coordinate.
- double getY ()
  Returns the current UTM Northing coordinate.
- double getZ ()
  Returns the current altitude.
- int getZone ()
  Returns the current UTM zone.
- int getHemi ()
  Returns the current hemisphere.
- void setX (double x)

Set the UTM Easting directly.

- void setY (double y)

  Set the UTM Northing directly.
- void setZ (double z)

  Set the altitude directly.
- void setZone (int zone)

  Set the UTM zone directly.
- void setHemi (int hemi)

  Set the hemisphere directly.
- tf::Vector3 asVector3 ()

  Return the current UTM coordinates as a tf::Vector3.
- tf::Vector3 operator- (UTMCoords &utm2)

  Subtract one UTMCoords class instance from another to get relative position vector.
- LatLon operator+ (tf::Vector3 &local_coords)

  Add a relative position vector to current coordinates, and convert the result to lat/lon.
- LatLon **operator+** (const tf::Vector3 &local_coords)

2.1 Detailed Description

Class to contain a UTM point.

This class represents a geographic point in Universal Transverse Mercator (UTM) coordinates. Operators are overloaded to make converting to and from lat/lon more convenient.

2.2 Constructor & Destructor Documentation

2.2.1 UTMCoords::UTMCoords (double *x*, double *y*, double *z*, int *zone*, int *hemi*) [inline]

Initialize UTM coordinates directly.

Parameters

in	*x*	UTM Easting coordinate in meters
in	*y*	UTM Northing coordinate in meters
in	*z*	Altitude in meters
in	*zone*	UTM zone number
in	*hemi*	Hemisphere (1 = northern, -1 = southern)

2.2.2 UTMCoords::UTMCoords (const sensor_msgs::NavSatFix & *fix*) [inline]

Initialize coordinates using the data in the ROS message 'sensor_msgs::NavSatFix'.

Parameters

in	*fix*	

2.2.3 UTMCoords::UTMCoords (LatLon *fix*) [inline]

Initialize coordinates using an instance of the LatLon class.

Parameters

in	*fix*	LatLon class instance

2.3 Member Function Documentation

2.3.1 int UTMCoords::getHemi ( ) [inline]

Returns the current hemisphere.

Returns

UTM hemisphere (1 = northern, -1 = southern)

2.3.2 double UTMCoords::getX ( ) [inline]

Returns the current UTM Easting coordinate.

Returns

UTM Easting in meters

2.3.3 double UTMCoords::getY ( ) [inline]

Returns the current UTM Northing coordinate.

Returns

UTM Northing in meters

2.3.4 double UTMCoords::getZ ( ) [inline]

Returns the current altitude.

Returns

Altitude in meters

2.3.5 int UTMCoords::getZone ( ) [inline]

Returns the current UTM zone.

Returns

UTM zone number (1 - 60)

2.3.6 LatLon UTMCoords::operator+ (tf::Vector3 & *local_coords*) [inline]

Add a relative position vector to current coordinates, and convert the result to lat/lon.

Parameters

local_coords	Relative position vector from current UTM coordinates.

Returns

LatLon class instance representing the absolute position of 'local_coords'

2.3.7 tf::Vector3 UTMCoords::operator- (UTMCoords & *utm2*) [inline]

Subtract one UTMCoords class instance from another to get relative position vector.

Returns

Relative position vector as a tf::Vector3

2.3.8 void UTMCoords::setHemi (int *hemi*) [inline]

Set the hemisphere directly.

Parameters

in	*hemi*	Hemisphere (1 = northern, -1 = southern)

2.3.9 void UTMCoords::setX (double *x*) [inline]

Set the UTM Easting directly.

Parameters

in	*x*	UTM Easting in meters

2.3.10 void UTMCoords::setY (double *y*) [inline]

Set the UTM Northing directly.

Parameters

in	*y*	UTM Easting in meters

2.3.11 void UTMCoords::setZ (double *z*) [inline]

Set the altitude directly.

Parameters

in	*z*	Altitude in meters

2.3.12 void UTMCoords::setZone (int *zone*) [inline]

Set the UTM zone directly.

Parameters

in	*zone*	UTM zone number (1 - 60)

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

- include/ugv_course_libs/gps_conv.h