

OpenSky Planetarium

Generated by Doxygen 1.8.6

Mon Jan 30 2017 18:36:17

Contents

1	Hierarchical Index	1
1.1	Class Hierarchy	1
2	Class Index	3
2.1	Class List	3
3	File Index	5
3.1	File List	5
4	Class Documentation	7
4.1	Calibrate Class Reference	7
4.1.1	Detailed Description	7
4.1.2	Constructor & Destructor Documentation	7
4.1.2.1	Calibrate	7
4.1.3	Member Function Documentation	8
4.1.3.1	autoRef_3	8
4.1.3.2	getECoords	8
4.1.3.3	getHCoords	8
4.1.3.4	isConfigured	8
4.1.3.5	setRef_1	8
4.1.3.6	setRef_2	9
4.1.3.7	setRef_3	9
4.1.3.8	setTime	9
4.2	LaserDev Class Reference	9
4.2.1	Detailed Description	10
4.2.2	Constructor & Destructor Documentation	10
4.2.2.1	LaserDev	10
4.2.2.2	~LaserDev	10
4.2.3	Member Function Documentation	11
4.2.3.1	debug_send	11
4.2.3.2	getPos	11
4.2.3.3	init	11

4.2.3.4	laserOff	11
4.2.3.5	laserOn	11
4.2.3.6	move	11
4.2.3.7	movx	11
4.2.3.8	movy	11
4.2.3.9	pos_received	11
4.2.3.10	processError	12
4.2.3.11	processTimeout	12
4.2.3.12	setPortName	12
4.2.3.13	sread	12
4.2.3.14	stop	12
4.3	OpenSkyPlanetarium Class Reference	12
4.3.1	Detailed Description	13
4.3.2	Constructor & Destructor Documentation	13
4.3.2.1	OpenSkyPlanetarium	13
4.3.2.2	~OpenSkyPlanetarium	13
4.3.3	Member Function Documentation	13
4.3.3.1	configureGui	13
4.3.3.2	deinit	13
4.3.3.3	getCallOrder	13
4.3.3.4	init	13
4.3.3.5	update	14
4.4	OpenSkyPlanetariumPluginInterface Class Reference	14
4.4.1	Detailed Description	14
4.4.2	Member Function Documentation	14
4.4.2.1	getPluginInfo	14
4.4.2.2	getStelModule	14
4.5	OSPMainDialog Class Reference	15
4.5.1	Detailed Description	16
4.5.2	Constructor & Destructor Documentation	16
4.5.2.1	OSPMainDialog	16
4.5.2.2	~OSPMainDialog	16
4.5.3	Member Function Documentation	16
4.5.3.1	arrow_released	16
4.5.3.2	comGOTO	16
4.5.3.3	compileScript	16
4.5.3.4	comPLAY	16
4.5.3.5	comTURN	17
4.5.3.6	comWAIT	17
4.5.3.7	createDialogContent	17

4.5.3.8	debug_received	17
4.5.3.9	downPressed	17
4.5.3.10	execScript	17
4.5.3.11	goTo	17
4.5.3.12	gt	18
4.5.3.13	initDevice	18
4.5.3.14	laser	18
4.5.3.15	laserToggled	18
4.5.3.16	leftPressed	18
4.5.3.17	lo	18
4.5.3.18	move	18
4.5.3.19	openScript	18
4.5.3.20	pause	19
4.5.3.21	pl	19
4.5.3.22	play	19
4.5.3.23	playAudio	19
4.5.3.24	playClicked	19
4.5.3.25	pos_received	19
4.5.3.26	retranslate	19
4.5.3.27	rightPressed	19
4.5.3.28	saveScript	19
4.5.3.29	selectDevice	20
4.5.3.30	setReference	20
4.5.3.31	setSignals	20
4.5.3.32	setVolume	20
4.5.3.33	showMessage	20
4.5.3.34	stop	20
4.5.3.35	stopClick	20
4.5.3.36	upPressed	20
4.5.3.37	waitforsec	20
4.5.3.38	wt	21
4.6	SerialCom Class Reference	21
4.6.1	Detailed Description	21
4.6.2	Constructor & Destructor Documentation	22
4.6.2.1	SerialCom	22
4.6.2.2	~SerialCom	22
4.6.3	Member Function Documentation	22
4.6.3.1	error	22
4.6.3.2	response	22
4.6.3.3	run	22

4.6.3.4	sendRequest	22
4.6.3.5	timeout	22
5	File Documentation	23
5.1	Calibrate.cpp File Reference	23
5.2	Calibrate.hpp File Reference	23
5.3	gui/OSPMainDialog.cpp File Reference	24
5.4	gui/OSPMainDialog.hpp File Reference	24
5.5	LaserDev.cpp File Reference	25
5.6	LaserDev.hpp File Reference	25
5.7	OpenSkyPlanetarium.cpp File Reference	25
5.8	OpenSkyPlanetarium.hpp File Reference	26
5.9	SerialCom.cpp File Reference	26
5.10	SerialCom.hpp File Reference	26
Index		27

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

Calibrate	7
QObject	
LaserDev	9
OpenSkyPlanetariumPluginInterface	14
QThread	
SerialCom	21
StelDialog	
OSPMainDialog	15
StelModule	
OpenSkyPlanetarium	12
StelPluginInterface	
OpenSkyPlanetariumPluginInterface	14

Chapter 2

Class Index

2.1 Class List

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

Calibrate	Library for coordinates transformations. Calculates the equivalent coordinates between both coordinate systems equatorial and horizontal	7
LaserDev	This class is used for funtions related to motion and turning the laser on and off	9
OpenSkyPlanetarium	This is used to dynamically load Open Sky Planetarium plugin into stellarium	12
OpenSkyPlanetariumPluginInterface	This class is used by Qt to manage a plug-in interface	14
OSPMainDialog	This is the main class used in connecting all the signals to gui	15
SerialCom	This class is used for funtions related to serial communication with the arduino	21

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

Calibrate.cpp	23
Calibrate.hpp	23
LaserDev.cpp	25
LaserDev.hpp	25
OpenSkyPlanetarium.cpp	25
OpenSkyPlanetarium.hpp	26
SerialCom.cpp	26
SerialCom.hpp	26
gui/OSPMainDialog.cpp	24
gui/OSPMainDialog.hpp	24

Chapter 4

Class Documentation

4.1 Calibrate Class Reference

Library for coordinates transformations. Calculates the equivalent coordinates between both coordinate systems equatorial and horizontal.

```
#include <Calibrate.hpp>
```

Public Member Functions

- [Calibrate](#) ()
- void [setTime](#) (double t0)
- void [setRef_1](#) (double ar, double dec, double t, double ac, double alt)
- void [setRef_2](#) (double ar, double dec, double t, double ac, double alt)
- void [setRef_3](#) (double ar, double dec, double t, double ac, double alt)
- void [autoRef_3](#) ()
- bool [isConfigured](#) ()
- void [getHCoords](#) (double ar, double dec, double t, double *ac, double *alt)
- void [getECoords](#) (double ac, double alt, double t, double *ar, double *dec)

4.1.1 Detailed Description

Library for coordinates transformations. Calculates the equivalent coordinates between both coordinate systems equatorial and horizontal.

It's based on Toshimi Taki's matrix method for coordinates transformation: http://www.geocities.jp/toshimi_taki/matrix/matrix.htm Contains the necessary methods for setting the initial time, the reference objects, the transformation matrix, and to calculate the equivalent vectors between both coordinate systems.

Definition at line 36 of file Calibrate.hpp.

4.1.2 Constructor & Destructor Documentation

4.1.2.1 Calibrate::Calibrate ()

Class constructor.

Definition at line 48 of file Calibrate.cpp.

4.1.3 Member Function Documentation

4.1.3.1 void Calibrate::autoRef_3 ()

Third reference object calculated from the two others ones.

Calculates the cross product of the two first reference objects in both coordinates systems, in order to obtain the third one. These two first objects must have 90° from each other, approximately (from 60° to 120° is enough to obtain goods results).

Definition at line 175 of file Calibrate.cpp.

4.1.3.2 void Calibrate::getECoords (double *ac*, double *alt*, double *t*, double * *ar*, double * *dec*)

Equatorial coordinates calculated from the horizontal ones and time.

Parameters

<i>ac</i>	Azimuth (horizontal coordinates).
<i>alt</i>	Altitude (horizontal coordinates).
<i>t</i>	Unix Timestamp of the Observation.
* <i>ar</i>	Pointer to double: Returns the right ascension (equatorial coordinates).
* <i>dec</i>	Pointer to double: Returns the declination (equatorial coordinates).

Definition at line 260 of file Calibrate.cpp.

4.1.3.3 void Calibrate::getHCoords (double *ar*, double *dec*, double *t*, double * *ac*, double * *alt*)

Horizontal coordinates calculated from the equatorial ones and time.

Parameters

<i>ar</i>	Right Ascension (equatorial coordinates).
<i>dec</i>	Declination (equatorial coordinates)
<i>t</i>	Unix Timestamp of the Observation.
* <i>ac</i>	Pointer to double: Returns the azimuth (horizontal coordiantes).
* <i>alt</i>	Pointer to double: Returns the altitude (horizontal coordinates).

Definition at line 231 of file Calibrate.cpp.

4.1.3.4 bool Calibrate::isConfigured ()

Indicates if the three reference objects has been calculated.

Returns

Boolean.

Definition at line 167 of file Calibrate.cpp.

4.1.3.5 void Calibrate::setRef_1 (double *ar*, double *dec*, double *t*, double *ac*, double *alt*)

Sets the first reference object from the coordinates in both coordinates systems for that object.

Parameters

<i>ar</i>	Right Ascension (equatorial coordinates).
<i>dec</i>	Declination (equatorial coordinates).
<i>t</i>	Unix Timestamp of the Observation.
<i>ac</i>	Azimuth (horizontal coordinates).
<i>alt</i>	Altitude (horizontal coordinates).

Definition at line 127 of file Calibrate.cpp.

4.1.3.6 void Calibrate::setRef_2 (double *ar*, double *dec*, double *t*, double *ac*, double *alt*)

Sets the second reference object from the coordinates in both coordinates systems for that object.

Parameters

<i>ar</i>	Right Ascension (equatorial coordinates).
<i>dec</i>	Declination (equatorial coordinates).
<i>t</i>	Unix Timestamp of the Observation.
<i>ac</i>	Azimuth (horizontal coordinates).
<i>alt</i>	Altitude (horizontal coordinates).

Definition at line 141 of file Calibrate.cpp.

4.1.3.7 void Calibrate::setRef_3 (double *ar*, double *dec*, double *t*, double *ac*, double *alt*)

Sets the third reference object from the coordinates in both coordinates systems for that object.

Parameters

<i>ar</i>	Right Ascension (equatorial coordinates).
<i>dec</i>	Declination (equatorial coordinates).
<i>t</i>	Unix Timestamp of the Observation.
<i>ac</i>	Azimuth (horizontal coordinates).
<i>alt</i>	Altitude (horizontal coordinates).

Definition at line 155 of file Calibrate.cpp.

4.1.3.8 void Calibrate::setTime (double *t0*)

Sets the initial time.

This parameter is used in order to consider time passing on horizontal coordinates system.

Parameters

<i>t0</i>	Unix Timestamp of the initial observation time.
-----------	---

Definition at line 119 of file Calibrate.cpp.

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

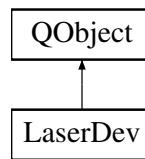
- [Calibrate.hpp](#)
- [Calibrate.cpp](#)

4.2 LaserDev Class Reference

This class is used for funtions related to motion and turning the laser on and off.

```
#include <LaserDev.hpp>
```

Inheritance diagram for LaserDev:



Public Slots

- void [getPos](#) ()
- void [processError](#) (const QString &s)
- void [processTimeout](#) (const QString &s)
- void [sread](#) (const QString &s)
- void [init](#) ()
- void [move](#) (double x, double y)
- void [movx](#) (int signDir)
- void [movy](#) (int signDir)
- void [stop](#) ()
- void [laserOn](#) ()
- void [laserOff](#) ()

Signals

- void [debug_send](#) (QString s)
- void [pos_received](#) (QString, QString)

Public Member Functions

- [LaserDev](#) (QObject *parent=0)
- [~LaserDev](#) ()
- void [setPortName](#) (const QString &s)

4.2.1 Detailed Description

This class is used for funtions related to motion and turning the laser on and off.

Definition at line 32 of file LaserDev.hpp.

4.2.2 Constructor & Destructor Documentation

4.2.2.1 LaserDev::LaserDev (QObject * *parent* = 0)

Class constructor.

Definition at line 35 of file LaserDev.cpp.

4.2.2.2 LaserDev::~~LaserDev ()

Class destructor.

Definition at line 48 of file LaserDev.cpp.

4.2.3 Member Function Documentation

4.2.3.1 void LaserDev::debug_send (QString s) [signal]

Sending the debug signal

4.2.3.2 void LaserDev::getPos () [slot]

This function is called to get the current position of the telescope in radians

Definition at line 151 of file LaserDev.cpp.

4.2.3.3 void LaserDev::init () [slot]

sends the "init" command to the device and after "done_init" is received from the device, emits "init_received" signal

Definition at line 141 of file LaserDev.cpp.

4.2.3.4 void LaserDev::laserOff () [slot]

To turn the laser off

Definition at line 222 of file LaserDev.cpp.

4.2.3.5 void LaserDev::laserOn () [slot]

to turn the laser on

Definition at line 212 of file LaserDev.cpp.

4.2.3.6 void LaserDev::move (double x, double y) [slot]

Sends the telescope coordinates to the device

Parameters

x	Azimuth fed to arduino.
y	Altitude fed to arduino.

Definition at line 162 of file LaserDev.cpp.

4.2.3.7 void LaserDev::movx (int signDir) [slot]

To move the laser in x direction depending on signDir

Definition at line 191 of file LaserDev.cpp.

4.2.3.8 void LaserDev::movy (int signDir) [slot]

To move the laser in y direction depending on signDir /param signDir 0 to move up 1 to move down

Definition at line 178 of file LaserDev.cpp.

4.2.3.9 void LaserDev::pos_received (QString , QString) [signal]

Receiving the position from arduino

4.2.3.10 void LaserDev::processError (const QString & s) [slot]

slot for the error, sends the signal to [OSPMainDialog](#)

Definition at line 56 of file LaserDev.cpp.

4.2.3.11 void LaserDev::processTimeout (const QString & s) [slot]

slot for the timeout , sends the signal to [OSPMainDialog](#)

Definition at line 63 of file LaserDev.cpp.

4.2.3.12 void LaserDev::setPortName (const QString & s)

function for setting the portName

Parameters

s	port name
----------	-----------

Definition at line 71 of file LaserDev.cpp.

4.2.3.13 void LaserDev::sread (const QString & s) [slot]

This function is called after writing to the serial port. This function performs various steps like echoData, getSteps, getHorizontalCoords, getVerticalCoords

Definition at line 81 of file LaserDev.cpp.

4.2.3.14 void LaserDev::stop () [slot]

to stop the laser movements in either of the direction x or y /param signDir 0 to move up 1 to move down

Definition at line 203 of file LaserDev.cpp.

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

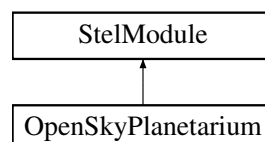
- [LaserDev.hpp](#)
- [LaserDev.cpp](#)

4.3 OpenSkyPlanetarium Class Reference

This is used to dynamically load Open Sky Planetarium plugin into stellarium.

```
#include <OpenSkyPlanetarium.hpp>
```

Inheritance diagram for OpenSkyPlanetarium:



Public Member Functions

- [OpenSkyPlanetarium \(\)](#)

- [~OpenSkyPlanetarium](#) ()
- virtual void [init](#) ()
- virtual void [deinit](#) ()
- virtual void [update](#) (double)
- virtual double [getCallOrder](#) (StelModuleActionName actionName) const
- virtual bool [configureGui](#) (bool show)

4.3.1 Detailed Description

This is used to dynamically load Open Sky Planetarium plugin into stellarium.

Definition at line 30 of file OpenSkyPlanetarium.hpp.

4.3.2 Constructor & Destructor Documentation

4.3.2.1 OpenSkyPlanetarium::OpenSkyPlanetarium ()

Class constructor.

Definition at line 60 of file OpenSkyPlanetarium.cpp.

4.3.2.2 OpenSkyPlanetarium::~~OpenSkyPlanetarium ()

Class destructor.

Definition at line 69 of file OpenSkyPlanetarium.cpp.

4.3.3 Member Function Documentation

4.3.3.1 bool OpenSkyPlanetarium::configureGui (bool *show*) [virtual]

This method is used to show the plugin interface

Definition at line 86 of file OpenSkyPlanetarium.cpp.

4.3.3.2 void OpenSkyPlanetarium::deinit () [virtual]

This method is used to delete the instance of main Window

Definition at line 122 of file OpenSkyPlanetarium.cpp.

4.3.3.3 double OpenSkyPlanetarium::getCallOrder (StelModuleActionName *actionName*) const [virtual]

This method is used to reimplement the getCallOrder method

Definition at line 76 of file OpenSkyPlanetarium.cpp.

4.3.3.4 void OpenSkyPlanetarium::init () [virtual]

Init Open Sky Planetarium module

Definition at line 94 of file OpenSkyPlanetarium.cpp.

4.3.3.5 `virtual void OpenSkyPlanetarium::update (double) [inline],[virtual]`

Definition at line 57 of file OpenSkyPlanetarium.hpp.

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

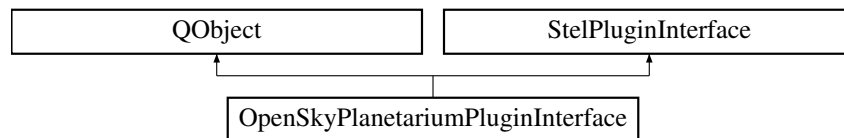
- [OpenSkyPlanetarium.hpp](#)
- [OpenSkyPlanetarium.cpp](#)

4.4 OpenSkyPlanetariumPluginInterface Class Reference

This class is used by Qt to manage a plug-in interface.

```
#include <OpenSkyPlanetarium.hpp>
```

Inheritance diagram for OpenSkyPlanetariumPluginInterface:



Public Member Functions

- virtual StelModule * [getStelModule](#) () const
- virtual StelPluginInfo [getPluginInfo](#) () const

4.4.1 Detailed Description

This class is used by Qt to manage a plug-in interface.

Definition at line 79 of file OpenSkyPlanetarium.hpp.

4.4.2 Member Function Documentation

4.4.2.1 `StelPluginInfo OpenSkyPlanetariumPluginInterface::getPluginInfo () const [virtual]`

This method is used to pass the information about the OpenSky Planetarium plugin.

Definition at line 43 of file OpenSkyPlanetarium.cpp.

4.4.2.2 `StelModule * OpenSkyPlanetariumPluginInterface::getStelModule () const [virtual]`

This method is the one called automatically by the StelModuleMgr just after loading the dynamic library

Definition at line 38 of file OpenSkyPlanetarium.cpp.

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

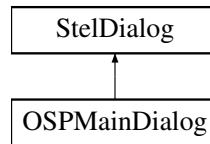
- [OpenSkyPlanetarium.hpp](#)
- [OpenSkyPlanetarium.cpp](#)

4.5 OSPMainDialog Class Reference

This is the main class used in connecting all the signals to gui.

```
#include <OSPMainDialog.hpp>
```

Inheritance diagram for OSPMainDialog:



Public Slots

- void [retranslate](#) ()
- void [debug_received](#) (QString s)
- void [pos_received](#) (QString x, QString y)
- void [selectDevice](#) ()
- void [initDevice](#) ()
- void [arrow_released](#) ()
- void [upPressed](#) ()
- void [downPressed](#) ()
- void [rightPressed](#) ()
- void [leftPressed](#) ()
- void [laserToggled](#) ()
- void [setReference](#) ()
- void [goTo](#) ()
- void [openScript](#) ()
- void [saveScript](#) ()
- void [execScript](#) ()
- void [compileScript](#) ()
- void [laser](#) (bool stat)
- void [playAudio](#) (QString fname)
- void [waitforsec](#) (int min, int sec)
- void [move](#) (QString, QString)
- void [gt](#) ()
- void [pl](#) ()
- void [lo](#) ()
- void [wt](#) ()
- void [setVolume](#) (int volume)
- void [playClicked](#) ()
- void [stopClick](#) ()

Signals

- void [comGOTO](#) (QString sra, QString sdec)
- void [comTURN](#) (bool stat)
- void [comWAIT](#) (int min, int sec)
- void [comPLAY](#) (QString fname)
- void [play](#) ()
- void [pause](#) ()
- void [stop](#) ()

Public Member Functions

- [OSPMainDialog](#) ()
- [~OSPMainDialog](#) ()
- void [setSignals](#) ()
- void [showMessage](#) (QString m)

Protected Member Functions

- void [createDialogContent](#) ()

4.5.1 Detailed Description

This is the main class used in connecting all the signals to gui.

Definition at line 39 of file OSPMainDialog.hpp.

4.5.2 Constructor & Destructor Documentation

4.5.2.1 OSPMainDialog::OSPMainDialog ()

Definition at line 50 of file OSPMainDialog.cpp.

4.5.2.2 OSPMainDialog::~~OSPMainDialog ()

Definition at line 78 of file OSPMainDialog.cpp.

4.5.3 Member Function Documentation

4.5.3.1 void OSPMainDialog::arrow_released () [slot]

This function is called when the Up(mvUp), Down(mvDown), Right(mvRight) and Left(mvLeft) button is released
This function sends the stop command to the arduino device to stop its movement along any of the four directions

Definition at line 273 of file OSPMainDialog.cpp.

4.5.3.2 void OSPMainDialog::comGOTO (QString *sra*, QString *sdec*) [signal]

This signal is connected to move of [OSPMainDialog](#) class

Parameters

<i>sra</i>	Right Ascension (equatorial coordinates).
<i>dec</i>	Declination (equatorial coordinates).

4.5.3.3 void OSPMainDialog::compileScript () [slot]

This function is used to compile script. This function maps the user script commands to the C++ functions

Definition at line 447 of file OSPMainDialog.cpp.

4.5.3.4 void OSPMainDialog::comPLAY (QString *fname*) [signal]

This signal is connected to playAudio of [OSPMainDialog](#) class

Parameters

<i>fname</i>	file name
--------------	-----------

4.5.3.5 void OSPMainDialog::comTURN (bool *stat*) [signal]

This signal is connected to laser of [OSPMainDialog](#) class

Parameters

<i>stat</i>	status of laser
-------------	-----------------

4.5.3.6 void OSPMainDialog::comWAIT (int *min*, int *sec*) [signal]

This signal is connected to waitforsec of [OSPMainDialog](#) class

Parameters

<i>min</i>	time in minute
<i>sec</i>	time in second

4.5.3.7 void OSPMainDialog::createDialogContent () [protected]

This function is used to create a dialog box and set the current index of the box

Definition at line 92 of file OSPMainDialog.cpp.

4.5.3.8 void OSPMainDialog::debug_received (QString *s*) [slot]

This funtion is connected to many signals for debugging purpose.

Parameters

<i>s</i>	Debug string
----------	--------------

Definition at line 173 of file OSPMainDialog.cpp.

4.5.3.9 void OSPMainDialog::downPressed () [slot]

This functions is called when the buttons Down(mvDown) is pressed

Definition at line 286 of file OSPMainDialog.cpp.

4.5.3.10 void OSPMainDialog::execScript () [slot]

This function is used to execute script. This function calls compile function before executing

Definition at line 414 of file OSPMainDialog.cpp.

4.5.3.11 void OSPMainDialog::goTo () [slot]

This function sends the coordinates from stellarium to device so that the laser could point the star. This function is enabled only after calibration is performed

Definition at line 348 of file OSPMainDialog.cpp.

4.5.3.12 void OSPMainDialog::gt () [slot]

This slot is connected to Goto Button of the Script Engine. Adds the goto command to your script

Definition at line 605 of file OSPMainDialog.cpp.

4.5.3.13 void OSPMainDialog::initDevice () [slot]

This function initiates the arduino device i-e: Counts the no. of steps and sets device's position. This function is connected to "Start Calibration" (startCal) button of the gui

Definition at line 222 of file OSPMainDialog.cpp.

4.5.3.14 void OSPMainDialog::laser (bool stat) [slot]

This is a slot for our script engine emit signal comTURN. This is used when playing the script

Definition at line 553 of file OSPMainDialog.cpp.

4.5.3.15 void OSPMainDialog::laserToggled () [slot]

This function is connected to the laser turnOn/turnOff radioButtons of the gui

Definition at line 303 of file OSPMainDialog.cpp.

4.5.3.16 void OSPMainDialog::leftPressed () [slot]

This function is called when the buttons Left(mvLeft) is pressed

Definition at line 294 of file OSPMainDialog.cpp.

4.5.3.17 void OSPMainDialog::lo () [slot]

This slot is connected to laser on/off button of the Script Engine. Adds the laser on/off command to your script

Definition at line 640 of file OSPMainDialog.cpp.

4.5.3.18 void OSPMainDialog::move (QString x, QString y) [slot]

This slot is connected to goto command from our script engine. It takes ra/dec of star as its parameters and converts them to move

Parameters

<i>sra</i>	Right Ascension (equatorial coordinates).
<i>dec</i>	Declination (equatorial coordinates).

Definition at line 537 of file OSPMainDialog.cpp.

4.5.3.19 void OSPMainDialog::openScript () [slot]

This function opens an existing file if present in the script directory of our module.

Definition at line 375 of file OSPMainDialog.cpp.

4.5.3.20 void OSPMainDialog::pause () [signal]

This signal is connected to pause of QMediaPlayer class

4.5.3.21 void OSPMainDialog::pl () [slot]

This slot is connected to Play Button of the Script Engine. Adds the play audio command to your script

Definition at line 624 of file OSPMainDialog.cpp.

4.5.3.22 void OSPMainDialog::play () [signal]

This signal is connected to play of QMediaPlayer class

4.5.3.23 void OSPMainDialog::playAudio (QString *fname*) [slot]

This function is used to play audio files. This is used to give play Audio functionality in our script

Definition at line 587 of file OSPMainDialog.cpp.

4.5.3.24 void OSPMainDialog::playClicked () [slot]

This slot is connected to play and pause button.

Definition at line 702 of file OSPMainDialog.cpp.

4.5.3.25 void OSPMainDialog::pos_received (QString *x*, QString *y*) [slot]

This slot is called when the laser device sends us the coordinates The coordinates are then used for setting the references in transformation matrix

Parameters

<i>x</i>	Azimuth in string.
<i>y</i>	Altitude in string.

Definition at line 182 of file OSPMainDialog.cpp.

4.5.3.26 void OSPMainDialog::retranslate () [slot]

This function retranslate the language of plugin.

Definition at line 83 of file OSPMainDialog.cpp.

4.5.3.27 void OSPMainDialog::rightPressed () [slot]

This functions is called when the buttons Right(mvRight) is pressed

Definition at line 290 of file OSPMainDialog.cpp.

4.5.3.28 void OSPMainDialog::saveScript () [slot]

This function is used to save the script

Definition at line 393 of file OSPMainDialog.cpp.

4.5.3.29 void OSPMainDialog::selectDevice () [slot]

This slot is connected to the selectDev button of the gui. This slot shows a list of connected device to select from after selection enables many buttons.

Definition at line 241 of file OSPMainDialog.cpp.

4.5.3.30 void OSPMainDialog::setReference () [slot]

This function sets three references for the matrix transformation basically it sends three references for matrix transformation to the device for its calculations

Definition at line 321 of file OSPMainDialog.cpp.

4.5.3.31 void OSPMainDialog::setSignals ()

This function connects the various gui signals to its corresponding slots. This function is called in the [createDialog-Content\(\)](#) function of this class.

Definition at line 106 of file OSPMainDialog.cpp.

4.5.3.32 void OSPMainDialog::setVolume (int *volume*) [slot]

This slot is connected to volume slider.

Parameters

<i>volume</i>	by slider
---------------	-----------

Definition at line 678 of file OSPMainDialog.cpp.

4.5.3.33 void OSPMainDialog::showMessage (QString *m*)

This function is called to display error/information messages

Definition at line 208 of file OSPMainDialog.cpp.

4.5.3.34 void OSPMainDialog::stop () [signal]

This signal is connected to stop of QMediaPlayer class

4.5.3.35 void OSPMainDialog::stopClick () [slot]

This slot is connected to stop button.

Definition at line 689 of file OSPMainDialog.cpp.

4.5.3.36 void OSPMainDialog::upPressed () [slot]

This functions is called when the buttons Up(mvUp) is pressed

Definition at line 282 of file OSPMainDialog.cpp.

4.5.3.37 void OSPMainDialog::waitforsec (int *min*, int *sec*) [slot]

This is used when playing the script. This is used to give wait functionality in the script

Parameters

<i>min</i>	time in minute
<i>sec</i>	time in second

Definition at line 571 of file OSPMainDialog.cpp.

4.5.3.38 void OSPMainDialog::wt () [slot]

This slot is connected to wait button of the Script Engine. Adds the wait command to your script

Definition at line 658 of file OSPMainDialog.cpp.

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

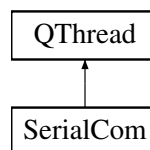
- [gui/OSPMainDialog.hpp](#)
- [gui/OSPMainDialog.cpp](#)

4.6 SerialCom Class Reference

This class is used for funtions related to serial communication with the arduino.

```
#include <SerialCom.hpp>
```

Inheritance diagram for SerialCom:



Signals

- void [response](#) (const QString &s)
- void [error](#) (const QString &s)
- void [timeout](#) (const QString &s)

Public Member Functions

- [SerialCom](#) (QObject *parent=0)
- [~SerialCom](#) ()
- void [sendRequest](#) (const QString &port, int waitTime, const QString &request)
- void [run](#) ()

4.6.1 Detailed Description

This class is used for funtions related to serial communication with the arduino.

Definition at line 26 of file SerialCom.hpp.

4.6.2 Constructor & Destructor Documentation

4.6.2.1 QT_USE_NAMESPACE SerialCom::SerialCom (QObject * *parent* = 0)

Class constructor

Constructor:

Definition at line 27 of file SerialCom.cpp.

4.6.2.2 SerialCom::~SerialCom ()

Class destructor

Destructor:

Definition at line 35 of file SerialCom.cpp.

4.6.3 Member Function Documentation

4.6.3.1 void SerialCom::error (const QString & *s*) [signal]

Error from arduino

4.6.3.2 void SerialCom::response (const QString & *s*) [signal]

Response from the arduino

4.6.3.3 void SerialCom::run ()

To send a request to arduino and wait for response and to save it

[run\(\)](#)

Definition at line 64 of file SerialCom.cpp.

4.6.3.4 void SerialCom::sendRequest (const QString & *port*, int *waitTime*, const QString & *request*)

To set the parameters that needed to sent to arduino

Parameters

<i>port</i>	name of the port on which request need to be sent
<i>waitTime</i>	wait time for the response from arduino
<i>request</i>	a string to be sent to arduino

sendRequest(const QString, int, const QString): Send a request to arduino

Definition at line 48 of file SerialCom.cpp.

4.6.3.5 void SerialCom::timeout (const QString & *s*) [signal]

Timeout error from arduino

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

- [SerialCom.hpp](#)
- [SerialCom.cpp](#)

Chapter 5

File Documentation

5.1 Calibrate.cpp File Reference

```
#include "Calibrate.hpp"
#include <QMediaPlayer>
#include <QCompleter>
#include <QFileDialog>
#include <QFile>
#include <QDebug>
#include <QTextStream>
#include <QObject>
#include <QString>
#include <QTime>
#include <QDateTime>
#include <QMessageBox>
#include <QInputDialog>
#include <QException>
#include <QRegExp>
#include <QRegExpValidator>
#include <QList>
#include <QWidget>
#include <QByteArray>
#include <QCoreApplication>
#include <stdio.h>
```

5.2 Calibrate.hpp File Reference

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

Classes

- class [Calibrate](#)

Library for coordinates transformations. Calculates the equivalent coordinates between both coordinate systems equatorial and horizontal.

5.3 gui/OSPMainDialog.cpp File Reference

```
#include "OpenSkyPlanetarium.hpp"
#include "OSPMainDialog.hpp"
#include "ui_OSPMainDialog.h"
#include "Calibrate.hpp"
#include "StelApp.hpp"
#include "StelLocaleMgr.hpp"
#include "StelModule.hpp"
#include "StelModuleMgr.hpp"
#include "StelObjectMgr.hpp"
#include "StelUtils.hpp"
#include "StelFileMgr.hpp"
#include "StarMgr.hpp"
#include "StelAudioMgr.hpp"
#include <QMediaPlayer>
#include <QCompleter>
#include <QFileDialog>
#include <QFile>
#include <QInputDialog>
#include <QException>
#include <QRegExp>
#include <QRegExpValidator>
#include <QList>
#include <QMessageBox>
#include <QWidget>
#include <QByteArray>
#include <QDateTime>
#include <QCoreApplication>
```

5.4 gui/OSPMainDialog.hpp File Reference

```
#include "Calibrate.hpp"
#include "StelDialog.hpp"
#include "LaserDev.hpp"
#include "StarMgr.hpp"
#include <QMediaPlayer>
#include <QWidget>
#include <QtSerialPort/QSerialPort>
#include <QtSerialPort/QSerialPortInfo>
#include <QList>
#include <QHash>
#include <QMediaPlaylist>
#include <QAbstractSlider>
```

Classes

- class [OSPMainDialog](#)

This is the main class used in connecting all the signals to gui.

5.5 LaserDev.cpp File Reference

```
#include "LaserDev.hpp"
#include <QtSerialPort/QSerialPortInfo>
#include <QtSerialPort/QSerialPort>
#include <QCoreApplication>
#include <QTextStream>
#include <QObject>
#include <QDebug>
#include <QString>
#include <QTime>
#include <QDateTime>
#include <QMessageBox>
#include "SerialCom.hpp"
```

5.6 LaserDev.hpp File Reference

```
#include <QtSerialPort/QSerialPortInfo>
#include <QtSerialPort/QSerialPort>
#include <QTextStream>
#include <QCoreApplication>
#include <QString>
#include <QObject>
#include <QThread>
#include <QTime>
#include "SerialCom.hpp"
```

Classes

- class [LaserDev](#)

This class is used for funtions related to motion and turning the laser on and off.

5.7 OpenSkyPlanetarium.cpp File Reference

```
#include "StelApp.hpp"
#include "StelCore.hpp"
#include "StelLocaleMgr.hpp"
#include "StelModuleMgr.hpp"
#include "StelGui.hpp"
#include "StelGuiItems.hpp"
#include "OpenSkyPlanetarium.hpp"
#include "OSPMainDialog.hpp"
#include <QAction>
#include <QString>
#include <QDebug>
#include <stdexcept>
```

5.8 OpenSkyPlanetarium.hpp File Reference

```
#include "StelModule.hpp"
#include "StelGui.hpp"
#include <QString>
#include <QFont>
#include <QObject>
#include "StelPluginInterface.hpp"
```

Classes

- class [OpenSkyPlanetarium](#)
This is used to dynamically load Open Sky Planetarium plugin into stellarium.
- class [OpenSkyPlanetariumPluginInterface](#)
This class is used by Qt to manage a plug-in interface.

5.9 SerialCom.cpp File Reference

```
#include "SerialCom.hpp"
#include <QtSerialPort/QSerialPort>
#include <QTime>
```

5.10 SerialCom.hpp File Reference

```
#include <QThread>
#include <QMutex>
#include <QWaitCondition>
```

Classes

- class [SerialCom](#)
This class is used for funtions related to serial communication with the arduino.

Index

- ~LaserDev
 - LaserDev, [10](#)
- ~OSPMainDialog
 - OSPMainDialog, [16](#)
- ~OpenSkyPlanetarium
 - OpenSkyPlanetarium, [13](#)
- ~SerialCom
 - SerialCom, [22](#)
- arrow_released
 - OSPMainDialog, [16](#)
- autoRef_3
 - Calibrate, [8](#)
- Calibrate, [7](#)
 - autoRef_3, [8](#)
 - Calibrate, [7](#)
 - getECoords, [8](#)
 - getHCoords, [8](#)
 - isConfigured, [8](#)
 - setRef_1, [8](#)
 - setRef_2, [9](#)
 - setRef_3, [9](#)
 - setTime, [9](#)
- Calibrate.cpp, [23](#)
- Calibrate.hpp, [23](#)
- comGOTO
 - OSPMainDialog, [16](#)
- comPLAY
 - OSPMainDialog, [16](#)
- comTURN
 - OSPMainDialog, [17](#)
- comWAIT
 - OSPMainDialog, [17](#)
- compileScript
 - OSPMainDialog, [16](#)
- configureGui
 - OpenSkyPlanetarium, [13](#)
- createDialogContent
 - OSPMainDialog, [17](#)
- debug_received
 - OSPMainDialog, [17](#)
- debug_send
 - LaserDev, [11](#)
- deinit
 - OpenSkyPlanetarium, [13](#)
- downPressed
 - OSPMainDialog, [17](#)
- error
 - SerialCom, [22](#)
- execScript
 - OSPMainDialog, [17](#)
- getCallOrder
 - OpenSkyPlanetarium, [13](#)
- getECoords
 - Calibrate, [8](#)
- getHCoords
 - Calibrate, [8](#)
- getPluginInfo
 - OpenSkyPlanetariumPluginInterface, [14](#)
- getPos
 - LaserDev, [11](#)
- getStelModule
 - OpenSkyPlanetariumPluginInterface, [14](#)
- goTo
 - OSPMainDialog, [17](#)
- gt
 - OSPMainDialog, [17](#)
- gui/OSPMainDialog.cpp, [24](#)
- gui/OSPMainDialog.hpp, [24](#)
- init
 - LaserDev, [11](#)
 - OpenSkyPlanetarium, [13](#)
- initDevice
 - OSPMainDialog, [18](#)
- isConfigured
 - Calibrate, [8](#)
- laser
 - OSPMainDialog, [18](#)
- LaserDev, [9](#)
 - ~LaserDev, [10](#)
 - debug_send, [11](#)
 - getPos, [11](#)
 - init, [11](#)
 - LaserDev, [10](#)
 - laserOff, [11](#)
 - laserOn, [11](#)
 - LaserDev, [10](#)
 - move, [11](#)
 - movx, [11](#)
 - movy, [11](#)
 - pos_received, [11](#)
 - processError, [11](#)
 - processTimeout, [12](#)
 - setPortName, [12](#)
 - sread, [12](#)

- stop, 12
- LaserDev.cpp, 25
- LaserDev.hpp, 25
- laserOff
 - LaserDev, 11
- laserOn
 - LaserDev, 11
- laserToggled
 - OSPMainDialog, 18
- leftPressed
 - OSPMainDialog, 18
- lo
 - OSPMainDialog, 18
- move
 - LaserDev, 11
 - OSPMainDialog, 18
- movx
 - LaserDev, 11
- movy
 - LaserDev, 11
- OSPMainDialog, 15
 - ~OSPMainDialog, 16
 - arrow_released, 16
 - comGOTO, 16
 - comPLAY, 16
 - comTURN, 17
 - comWAIT, 17
 - compileScript, 16
 - createDialogContent, 17
 - debug_received, 17
 - downPressed, 17
 - execScript, 17
 - goTo, 17
 - gt, 17
 - initDevice, 18
 - laser, 18
 - laserToggled, 18
 - leftPressed, 18
 - lo, 18
 - move, 18
 - OSPMainDialog, 16
 - openScript, 18
 - OSPMainDialog, 16
 - pause, 18
 - pl, 19
 - play, 19
 - playAudio, 19
 - playClicked, 19
 - pos_received, 19
 - retranslate, 19
 - rightPressed, 19
 - saveScript, 19
 - selectDevice, 19
 - setReference, 20
 - setSignals, 20
 - setVolume, 20
 - showMessage, 20
 - stop, 20
 - stopClick, 20
 - upPressed, 20
 - waitforsec, 20
 - wt, 21
- openScript
 - OSPMainDialog, 18
- OpenSkyPlanetarium, 12
 - ~OpenSkyPlanetarium, 13
 - configureGui, 13
 - deinit, 13
 - getCallOrder, 13
 - init, 13
 - OpenSkyPlanetarium, 13
 - OpenSkyPlanetarium, 13
 - update, 13
- OpenSkyPlanetarium.cpp, 25
- OpenSkyPlanetarium.hpp, 26
- OpenSkyPlanetariumPluginInterface, 14
 - getPluginInfo, 14
 - getStelModule, 14
- pause
 - OSPMainDialog, 18
- pl
 - OSPMainDialog, 19
- play
 - OSPMainDialog, 19
- playAudio
 - OSPMainDialog, 19
- playClicked
 - OSPMainDialog, 19
- pos_received
 - LaserDev, 11
 - OSPMainDialog, 19
- processError
 - LaserDev, 11
- processTimeout
 - LaserDev, 12
- response
 - SerialCom, 22
- retranslate
 - OSPMainDialog, 19
- rightPressed
 - OSPMainDialog, 19
- run
 - SerialCom, 22
- saveScript
 - OSPMainDialog, 19
- selectDevice
 - OSPMainDialog, 19
- sendRequest
 - SerialCom, 22
- SerialCom, 21
 - ~SerialCom, 22
 - error, 22
 - response, 22

- run, [22](#)
- sendRequest, [22](#)
- SerialCom, [22](#)
- SerialCom, [22](#)
- timeout, [22](#)
- SerialCom.cpp, [26](#)
- SerialCom.hpp, [26](#)
- setPortName
 - LaserDev, [12](#)
- setRef_1
 - Calibrate, [8](#)
- setRef_2
 - Calibrate, [9](#)
- setRef_3
 - Calibrate, [9](#)
- setReference
 - OSPMainDialog, [20](#)
- setSignals
 - OSPMainDialog, [20](#)
- setTime
 - Calibrate, [9](#)
- setVolume
 - OSPMainDialog, [20](#)
- showMessage
 - OSPMainDialog, [20](#)
- sread
 - LaserDev, [12](#)
- stop
 - LaserDev, [12](#)
 - OSPMainDialog, [20](#)
- stopClick
 - OSPMainDialog, [20](#)
- timeout
 - SerialCom, [22](#)
- upPressed
 - OSPMainDialog, [20](#)
- update
 - OpenSkyPlanetarium, [13](#)
- waitforsec
 - OSPMainDialog, [20](#)
- wt
 - OSPMainDialog, [21](#)