Third Assignment for the Experimental Robotics Laboratory course

Generated by Doxygen 1.8.17

Chapter 1

Namespace Index

1.1 Packages

Here are the packages with brief descriptions (if available):

human . . . . . ??
sensor . . . . . ??
state_machine . . . . . ??

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

State
- state_machine.Find . . . . . **??**
- state_machine.Normal . . . . . **??**
- state_machine.Play . . . . . **??**
- state_machine.Sleep . . . . . **??**

Chapter 3

Class Index

3.1 Class List

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

state_machine.Find
    Define Find state . . . . . ??
state_machine.Normal
    Define Normal state . . . . . ??
state_machine.Play
    Define Play state . . . . . ??
state_machine.Sleep
    Define Sleep state . . . . . ??

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

scripts/human.py . . . . . ??
scripts/sensor.py . . . . . ??
scripts/state_machine.py . . . . . ??

Chapter 5

Namespace Documentation

5.1 human Namespace Reference

Functions

- def sendPlayCommand ()

  Publishes a simple "Play" command on the topic.
- def sendGoToCommand ()

  Publishes a "GoTo: Location" command on the topic by choosing randomly the location.
- def human ()

  Randomly execute one of the two "send command" functions.

Variables

- comPub = None

  Publisher.
- command = String()

  Command.
- logfile = None

  Log file.
- script_path = os.path.abspath(__file__)
- path_list = script_path.split(os.sep)
- script_directory = path_list[0:len(path_list)-2]
- string file_path = "log/human_logfile.txt"
- string path = "/".join(script_directory) + "/" + file_path

5.1.1 Detailed Description

Implements the basic behaviour of a person controlling the robot via voice commands. The person keeps sending either "Play" or "GoTo: Location" commands randomly.

5.1.2 Function Documentation

5.1.2.1 human()

```
def human.human ( )
```

Randomly execute one of the two "send command" functions.

5.1.2.2 sendGoToCommand()

```
def human.sendGoToCommand ( )
```

Publishes a "GoTo: Location" command on the topic by choosing randomly the location.

5.1.2.3 sendPlayCommand()

```
def human.sendPlayCommand ( )
```

Publishes a simple "Play" command on the topic.

5.1.3 Variable Documentation

5.1.3.1 command

```
human.command = String()
```

Command.

5.1.3.2 comPub

```
human.comPub = None
```

Publisher.

5.1.3.3 file_path

```
string human.file_path = "log/human_logfile.txt"
```

5.1.3.4 logfile

```
human.logfile = None
```

Log file.

5.1.3.5 path

```
string human.path = "/".join(script_directory) + "/" + file_path
```

5.1.3.6 path_list

```
human.path_list = script_path.split(os.sep)
```

5.1.3.7 script_directory

```
human.script_directory = path_list[0:len(path_list)-2]
```

5.1.3.8 script_path

```
human.script_path = os.path.abspath(__file__)
```

5.2 sensor Namespace Reference

Functions

- def formatCommand (ros_data)

  Understands the command told by the user and format it to send it to the robot.
- def formatPlay (command)

  Formats a "Play" command.
- def formatGoTo (command)

  Formats a "GoTo" command.

Variables

- formComPub = None

  Publisher.
- comSub = None

  Subscriber.
- formCommand = FormattedCommand()

  Formatted command.
- logfile = None

  Log file.
- script_path = os.path.abspath(__file__)
- path_list = script_path.split(os.sep)
- script_directory = path_list[0:len(path_list)-2]
- string file_path = "log/sensor_logfile.txt"
- string path = "/".join(script_directory) + "/" + file_path

5.2.1 Detailed Description

Acts as a robot sensor that perceives the commands told by the user. If a command is recognized, it's formatted and sent to the robot.

5.2.2 Function Documentation

5.2.2.1 formatCommand()

```
def sensor.formatCommand (
              ros_data )
```

Understands the command told by the user and format it to send it to the robot.

Parameters

ros_data	The unformatted command told by the user.

5.2.2.2 formatGoTo()

```
def sensor.formatGoTo (
              command )
```

Formats a "GoTo" command.

Parameters

command	The string list containing the command.

5.2.2.3 formatPlay()

```
def sensor.formatPlay (
              command )
```

Formats a "Play" command.

Parameters

command	The string list containing the command.

5.2.3 Variable Documentation

5.2.3.1 comSub

```
sensor.comSub = None
```

Subscriber.

5.2.3.2 file_path

```
string sensor.file_path = "log/sensor_logfile.txt"
```

5.2.3.3 formCommand

```
sensor.formCommand = FormattedCommand()
```

Formatted command.

5.2.3.4 formComPub

```
sensor.formComPub = None
```

Publisher.

5.2.3.5 logfile

```
sensor.logfile = None
```

Log file.

5.2.3.6 path

```
string sensor.path = "/".join(script_directory) + "/" + file_path
```

5.2.3.7 path_list

```
sensor.path_list = script_path.split(os.sep)
```

5.2.3.8 script_directory

```
sensor.script_directory = path_list[0:len(path_list)-2]
```

5.2.3.9 script_path

```
sensor.script_path = os.path.abspath(__file__)
```

5.3 state_machine Namespace Reference

Classes

- class Find

  Define Find state.
- class Normal

  Define Normal state.
- class Play

  Define Play state.
- class Sleep

  Define Sleep state.

Functions

- def checkContours (image, maskLower, maskUpper)

  Computes the contours of colored objects eventually present in the image.
- def checkForBall (ros_data)

  Checks if there's a ball and eventually moves the robot closer to it (TRACK sub-state) to save the corresponding location's position.
- def saveLocationPosition (locationNumber)

  Stores the position of a location.
- def updateRobotPosition (ros_data)

  Updates the current robot position in the plane.
- def receivedCommand (ros_data)

  Notifies that the state machine received a formatted command by raising the correct flags.
- def sendGoalNormalState ()

  Sends a random position goal to move_base for the NORMAL state and wait for its result.
- def sendGoalPlayState (x, y)

  Sends a goal to the move_base for the PLAY state and waits for its result.
- def startExploreLite ()

  Starts the explore-lite package.
- def stopExploreLite ()

  Stops the explore-lite package.
- def main ()

Variables

- movebaseClient = None

  Action client.
- velPub = None

  Publisher.
- imageSub = None

  Subscriber.
- odomSub = None

  Subscriber.
- commandSub = None

  Subscriber.
- mbGoal = MoveBaseGoal()

  Goal pose.
- int sleepCounter = 0

  Counter.
- bool receivedPlay = False

  Command variable.
- parameter = None

  Command variable.
- requestedLocation = None

  Command variable.
- bool ballFound = False

  Flag to notify that the robot has seen the ball.
- finishedMoving = threading.Event()

  Threading event.
- finishedGettingClose = threading.Event()

Threading event.

- stopExploring = threading.Event()

  Threading event.

- receivedGoTo = threading.Event()

  Threading event.

- tuple blueLower = (100, 50, 50)
- tuple blueUpper = (130, 255, 255)
- tuple redLower = (0, 225, 50)
- tuple redUpper = (5, 255, 255)
- tuple greenLower = (50, 50, 50)
- tuple greenUpper = (70, 255, 255)
- tuple yellowLower = (25, 50, 50)
- tuple yellowUpper = (35, 255, 255)
- tuple magentaLower = (135, 150, 50)
- tuple magentaUpper = (150, 255, 255)
- tuple blackLower = (0, 0, 0)
- tuple blackUpper = (5, 50, 50)
- list savedLocations = [ ]

  Saved locations list.

- robotPosition_x = None

  Current robot's x position.

- robotPosition_y = None

  Current robot's y position.

- logfile = None

  Log file.

5.3.1 Detailed Description

Defines the different robot behaviours and the transitions between them. Available states are NORMAL, SLEEP, PLAY and FIND.

5.3.2 Function Documentation

5.3.2.1 checkContours()

```
def state_machine.checkContours (
              image,
              maskLower,
              maskUpper )
```

Computes the contours of colored objects eventually present in the image.

Parameters

image	The preprocessed image in HSV format.
maskLower	The lower bound mask of the color of interest.
maskUpper	The upper bound mask of the color of interest.

5.3.2.2 checkForBall()

```
def state_machine.checkForBall (
              ros_data )
```

Checks if there's a ball and eventually moves the robot closer to it (TRACK sub-state) to save the corresponding location's position.

Parameters

ros_data	The compressed image picked up by the camera.

5.3.2.3 main()

```
def state_machine.main ( )
```

5.3.2.4 receivedCommand()

```
def state_machine.receivedCommand (
              ros_data )
```

Notifies that the state machine received a formatted command by raising the correct flags.

Parameters

ros_data	The received formatted command.

5.3.2.5 saveLocationPosition()

```
def state_machine.saveLocationPosition (
              locationNumber )
```

Stores the position of a location.

Parameters

locationNumber	The integer corresponding to a location.

5.3.2.6 sendGoalNormalState()

```
def state_machine.sendGoalNormalState ( )
```

Sends a random position goal to move_base for the NORMAL state and wait for its result.

5.3.2.7 sendGoalPlayState()

```
def state_machine.sendGoalPlayState (
              x,
              y )
```

Sends a goal to the move_base for the PLAY state and waits for its result.

Parameters

x	The x-coordinate of the location to go to.
y	The y-coordinate of the location to go to.

5.3.2.8 startExploreLite()

```
def state_machine.startExploreLite ( )
```

Starts the explore-lite package.

5.3.2.9 stopExploreLite()

```
def state_machine.stopExploreLite ( )
```

Stops the explore-lite package.

5.3.2.10 updateRobotPosition()

```
def state_machine.updateRobotPosition (
              ros_data )
```

Updates the current robot position in the plane.

Parameters

ros_data	The odometry information.

5.3.3 Variable Documentation

5.3.3.1 ballFound

```
bool state_machine.ballFound = False
```

Flag to notify that the robot has seen the ball.

5.3.3.2 blackLower

```
tuple state_machine.blackLower = (0, 0, 0)
```

5.3.3.3 blackUpper

```
tuple state_machine.blackUpper = (5, 50, 50)
```

5.3.3.4 blueLower

```
tuple state_machine.blueLower = (100, 50, 50)
```

5.3.3.5 blueUpper

```
tuple state_machine.blueUpper = (130, 255, 255)
```

5.3.3.6 commandSub

```
state_machine.commandSub = None
```

Subscriber.

5.3.3.7 finishedGettingClose

```
state_machine.finishedGettingClose = threading.Event()
```

Threading event.

5.3.3.8 finishedMoving

```
state_machine.finishedMoving = threading.Event()
```

Threading event.

5.3.3.9 greenLower

```
tuple state_machine.greenLower = (50, 50, 50)
```

5.3.3.10 greenUpper

```
tuple state_machine.greenUpper = (70, 255, 255)
```

5.3.3.11 imageSub

```
state_machine.imageSub = None
```

Subscriber.

5.3.3.12 logfile

```
state_machine.logfile = None
```

Log file.

5.3.3.13 magentaLower

```
tuple state_machine.magentaLower = (135, 150, 50)
```

5.3.3.14 magentaUpper

```
tuple state_machine.magentaUpper = (150, 255, 255)
```

5.3.3.15 mbGoal

```
state_machine.mbGoal = MoveBaseGoal()
```

Goal pose.

5.3.3.16 movebaseClient

```
state_machine.movebaseClient = None
```

Action client.

5.3.3.17 odomSub

```
state_machine.odomSub = None
```

Subscriber.

5.3.3.18 parameter

```
state_machine.parameter = None
```

Command variable.

5.3.3.19 receivedGoTo

```
state_machine.receivedGoTo = threading.Event()
```

Threading event.

5.3.3.20 receivedPlay

```
bool state_machine.receivedPlay = False
```

Command variable.

5.3.3.21 redLower

```
tuple state_machine.redLower = (0, 225, 50)
```

5.3.3.22 redUpper

```
tuple state_machine.redUpper = (5, 255, 255)
```

5.3.3.23 requestedLocation

```
state_machine.requestedLocation = None
```

Command variable.

5.3.3.24 robotPosition_x

```
state_machine.robotPosition_x = None
```

Current robot's x position.

5.3.3.25 robotPosition_y

```
state_machine.robotPosition_y = None
```

Current robot's y position.

5.3.3.26 savedLocations

```
list state_machine.savedLocations = []
```

Saved locations list.

5.3.3.27 sleepCounter

```
int state_machine.sleepCounter = 0
```

Counter.

5.3.3.28 stopExploring

```
state_machine.stopExploring = threading.Event()
```

Threading event.

5.3.3.29 velPub

```
state_machine.velPub = None
```

Publisher.

5.3.3.30 yellowLower

```
tuple state_machine.yellowLower = (25, 50, 50)
```

5.3.3.31 yellowUpper

```
tuple state_machine.yellowUpper = (35, 255, 255)
```

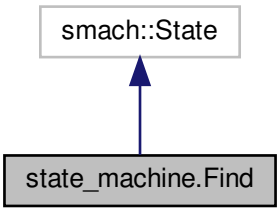
Chapter 6

Class Documentation

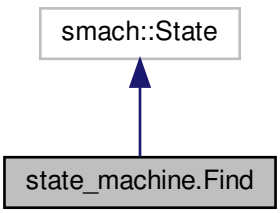
6.1 state_machine.Find Class Reference

Define Find state.

Inheritance diagram for state_machine.Find:



Collaboration diagram for state_machine.Find:



Public Member Functions

- def __init__ (self)
- def execute (self, userdata)

6.1.1 Detailed Description

Define Find state.

6.1.2 Constructor & Destructor Documentation

6.1.2.1 __init__()

```
def state_machine.Find.__init__ (
              self )
```

6.1.3 Member Function Documentation

6.1.3.1 execute()

```
def state_machine.Find.execute (
              self,
              userdata )
```

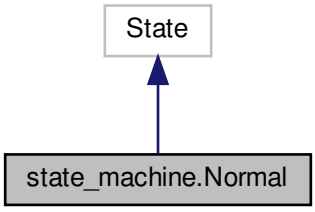
The documentation for this class was generated from the following file:

- scripts/state_machine.py

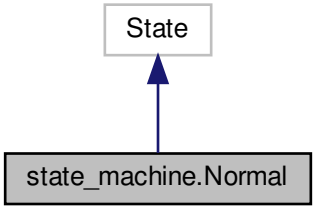
6.2 state_machine.Normal Class Reference

Define Normal state.

Inheritance diagram for state_machine.Normal:



Collaboration diagram for state_machine.Normal:



Public Member Functions

- def __init__ (self)
- def execute (self, userdata)

Public Attributes

- sleepThreshold

6.2.1 Detailed Description

Define Normal state.

6.2.2 Constructor & Destructor Documentation

6.2.2.1 __init__()

```
def state_machine.Normal.__init__ (
              self )
```

6.2.3 Member Function Documentation

6.2.3.1 execute()

```
def state_machine.Normal.execute (
              self,
              userdata )
```

6.2.4 Member Data Documentation

6.2.4.1 sleepThreshold

```
state_machine.Normal.sleepThreshold
```

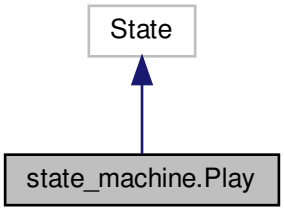
The documentation for this class was generated from the following file:

- scripts/state_machine.py

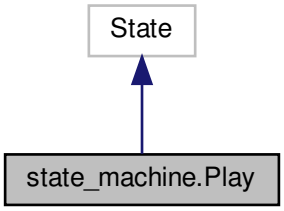
6.3 state_machine.Play Class Reference

Define Play state.

Inheritance diagram for state_machine.Play:



Collaboration diagram for state_machine.Play:



Public Member Functions

- def __init__ (self)
- def execute (self, userdata)

6.3.1 Detailed Description

Define Play state.

6.3.2 Constructor & Destructor Documentation

6.3.2.1 __init__()

```
def state_machine.Play.__init__ (
              self )
```

6.3.3 Member Function Documentation

6.3.3.1 execute()

```
def state_machine.Play.execute (
              self,
              userdata )
```

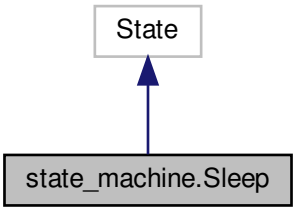
The documentation for this class was generated from the following file:

- scripts/state_machine.py

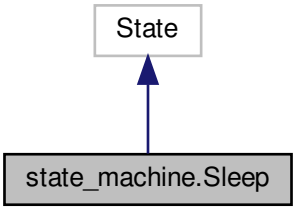
6.4 state_machine.Sleep Class Reference

Define Sleep state.

Inheritance diagram for state_machine.Sleep:



Collaboration diagram for state_machine.Sleep:



Public Member Functions

- def __init__ (self)
- def execute (self, userdata)

6.4.1 Detailed Description

Define Sleep state.

6.4.2 Constructor & Destructor Documentation

6.4.2.1 __init__()

```
def state_machine.Sleep.__init__ (
              self )
```

6.4.3 Member Function Documentation

6.4.3.1 execute()

```
def state_machine.Sleep.execute (
              self,
              userdata )
```

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

- scripts/state_machine.py

Chapter 7

File Documentation

7.1 scripts/human.py File Reference

Namespaces

- human

Functions

- def human.sendPlayCommand ()

  Publishes a simple "Play" command on the topic.
- def human.sendGoToCommand ()

  Publishes a "GoTo: Location" command on the topic by choosing randomly the location.
- def human.human ()

  Randomly execute one of the two "send command" functions.

Variables

- human.comPub = None

  Publisher.
- human.command = String()

  Command.
- human.logfile = None

  Log file.
- human.script_path = os.path.abspath(__file__)
- human.path_list = script_path.split(os.sep)
- human.script_directory = path_list[0:len(path_list)-2]
- string human.file_path = "log/human_logfile.txt"
- string human.path = "/".join(script_directory) + "/" + file_path

7.2 scripts/sensor.py File Reference

Namespaces

- sensor

Functions

- def sensor.formatCommand (ros_data)

  Understands the command told by the user and format it to send it to the robot.
- def sensor.formatPlay (command)

  Formats a "Play" command.
- def sensor.formatGoTo (command)

  Formats a "GoTo" command.

Variables

- sensor.formComPub = None

  Publisher.
- sensor.comSub = None

  Subscriber.
- sensor.formCommand = FormattedCommand()

  Formatted command.
- sensor.logfile = None

  Log file.
- sensor.script_path = os.path.abspath(__file__)
- sensor.path_list = script_path.split(os.sep)
- sensor.script_directory = path_list[0:len(path_list)-2]
- string sensor.file_path = "log/sensor_logfile.txt"
- string sensor.path = "/".join(script_directory) + "/" + file_path

7.3 scripts/state_machine.py File Reference

Classes

- class state_machine.Normal

  Define Normal state.
- class state_machine.Sleep

  Define Sleep state.
- class state_machine.Play

  Define Play state.
- class state_machine.Find

  Define Find state.

Namespaces

- state_machine

Functions

- def state_machine.checkContours (image, maskLower, maskUpper)

  Computes the contours of colored objects eventually present in the image.
- def state_machine.checkForBall (ros_data)

  Checks if there's a ball and eventually moves the robot closer to it (TRACK sub-state) to save the corresponding location's position.
- def state_machine.saveLocationPosition (locationNumber)

  Stores the position of a location.
- def state_machine.updateRobotPosition (ros_data)

  Updates the current robot position in the plane.
- def state_machine.receivedCommand (ros_data)

  Notifies that the state machine received a formatted command by raising the correct flags.
- def state_machine.sendGoalNormalState ()

  Sends a random position goal to move_base for the NORMAL state and wait for its result.
- def state_machine.sendGoalPlayState (x, y)

  Sends a goal to the move_base for the PLAY state and waits for its result.
- def state_machine.startExploreLite ()

  Starts the explore-lite package.
- def state_machine.stopExploreLite ()

  Stops the explore-lite package.
- def state_machine.main ()

Variables

- state_machine.movebaseClient = None

  Action client.
- state_machine.velPub = None

  Publisher.
- state_machine.imageSub = None

  Subscriber.
- state_machine.odomSub = None

  Subscriber.
- state_machine.commandSub = None

  Subscriber.
- state_machine.mbGoal = MoveBaseGoal()

  Goal pose.
- int state_machine.sleepCounter = 0

  Counter.
- bool state_machine.receivedPlay = False

  Command variable.
- state_machine.parameter = None

  Command variable.
- state_machine.requestedLocation = None

  Command variable.
- bool state_machine.ballFound = False

  Flag to notify that the robot has seen the ball.
- state_machine.finishedMoving = threading.Event()

  Threading event.
- state_machine.finishedGettingClose = threading.Event()

Threading event.

- state_machine.stopExploring = threading.Event()

  Threading event.

- state_machine.receivedGoTo = threading.Event()

  Threading event.

- tuple state_machine.blueLower = (100, 50, 50)
- tuple state_machine.blueUpper = (130, 255, 255)
- tuple state_machine.redLower = (0, 225, 50)
- tuple state_machine.redUpper = (5, 255, 255)
- tuple state_machine.greenLower = (50, 50, 50)
- tuple state_machine.greenUpper = (70, 255, 255)
- tuple state_machine.yellowLower = (25, 50, 50)
- tuple state_machine.yellowUpper = (35, 255, 255)
- tuple state_machine.magentaLower = (135, 150, 50)
- tuple state_machine.magentaUpper = (150, 255, 255)
- tuple state_machine.blackLower = (0, 0, 0)
- tuple state_machine.blackUpper = (5, 50, 50)
- list state_machine.savedLocations = [ ]

  Saved locations list.

- state_machine.robotPosition_x = None

  Current robot's x position.

- state_machine.robotPosition_y = None

  Current robot's y position.

- state_machine.logfile = None

  Log file.