Lego ev3 Fire Alarm Robot

Generated by Doxygen 1.9.8

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

Chapter 3

Class Index

3.1 Class List

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

Chapter 4

Namespace Documentation

4.1 behaviors Namespace Reference

Classes

- class FireDetection
- class RobotBehavior
- class TouchBehavior
- class WallFollowing
- class Wander

4.1.1 Detailed Description

```
responsible for coordinating the behaviors that the robot will take given the sensor data
```

4.2 globals Namespace Reference

Variables

- float **TILE_ERROR_DISTANCE** = 1.00

  COMPILE TIME CONSTANTS ##################### error factor when moving on surface.
- float **DISTANCE_ERROR** = TILE_ERROR_DISTANCE
- int **TILE_ERROR_TURN** = .950
- int **TURN_ERROR** = TILE_ERROR_TURN
- int **TIRE_CIRC** = 178

  ROBOT CONSTANTS ##################################.
- int **FULL_ROTATION** = 360
- int **TIRE_RPM** = 280
- int **ROBOT_LENGTH** = 105
- float **DIST_BTWN_WHEELS** = 158.0000
- tuple **ROBOT_RADIUS** = (DIST_BTWN_WHEELS/2.0000)
- float **M_PI** = 3.14159265359
- int **TOUCH** = 0
- int **FIRE** = 1
- int **WALL_FOLLOW** = 2
- int **WANDER** = 3
- int **MIN_WALL_DISTANCE** = 150

  ALARM CONSTANTS ##########################.
- int **BACKUP_DISTANCE** = -200
- int **ANGLE_LOW_BOUND** = 45
- int **ANGLE_UPPER_BOUND** = 180
- int **WALL_DISTANCE** = 30

4.2.1 Detailed Description

```
Contains constants and varaibles that will be modified before compile time
```

4.3 logger Namespace Reference

Functions

- None log (message, to_console_only=None)

4.3.1 Detailed Description

```
responsible for logging data to log.txt on lego ev3 pybrick
```

4.3.2 Function Documentation

4.3.2.1 log()

```
None logger.log (
              message,
              to_console_only = None )
```

```
Logs messages to log.txt on robot and prints to console as well\n
Args:
    message (str): msg to log
    to_console_only (Bool, optional): True if you want
    to print message only to the console. Defaults to None.
```

4.4 robotics Namespace Reference

Classes

- class Navigator
- class Robot

4.4.1 Detailed Description

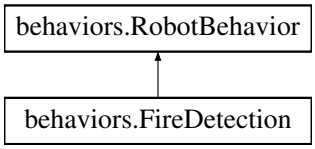
```
Acts as a replacement for pybricks robotics module. Module is
responsible for moving the robot, angle tracking, and robot behavior proccesing
```

Chapter 5

Class Documentation

5.1 behaviors.FireDetection Class Reference

Inheritance diagram for behaviors.FireDetection:



Public Member Functions

- __init__ (self)
- None run (self, robot)
- stop_behavior (self, robot, msg)
- None move_to_fire (self, robot)

Public Member Functions inherited from behaviors.RobotBehavior

- **__lt__** (self, other)

Additional Inherited Members

Public Attributes inherited from behaviors.RobotBehavior

- **priority**

5.1.1 Detailed Description

```
Coordinates robot behavior when it detects a red color in front of sensor. Priority of 1
```

5.1.2 Constructor & Destructor Documentation

5.1.2.1 __init__()

```
behaviors.FireDetection.__init__ (
              self )
```

Reimplemented from behaviors.RobotBehavior.

5.1.3 Member Function Documentation

5.1.3.1 move_to_fire()

```
None behaviors.FireDetection.move_to_fire (
              self,
              robot )
```

```
Verifies if the robot has found a fire. If the fire is found, sets fireNotFound to False.\n
Args: robot (Robot): the robot object
```

5.1.3.2 run()

```
None behaviors.FireDetection.run (
              self,
              robot )
```

```
Starts a behavior and executes their corresponding functions\n
Args: robot (Robot): the robot object
```

Reimplemented from behaviors.RobotBehavior.

5.1.3.3 stop_behavior()

```
behaviors.FireDetection.stop_behavior (
              self,
              robot,
              msg )
```

```
Stops the current behavior and resets the neccessary variables\n
Args: robot (Robot): the robot object
```

Reimplemented from behaviors.RobotBehavior.

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

- behaviors.py

5.2 robotics.Navigator Class Reference

Public Member Functions

- **__init__** (self)
- None update_nav (self, angle)

Public Attributes

- **orientation**
- **orientations**

5.2.1 Detailed Description

```
Class responsible for keeping track of the robot's logical orientation
```

5.2.2 Member Function Documentation

5.2.2.1 update_nav()

```
None robotics.Navigator.update_nav (
            self,
            angle )
```

```
Updates the logical orientation of the robot and keeps track of previous positions\n
Args: angle (int): angle that the robot needs to be be turned by
```

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

- robotics.py

5.3 robotics.Robot Class Reference

Public Member Functions

- **__init__** (self, Motor left_motor, Motor right_motor, Navigator navigator, TouchSensor frontTouch, Touch↩ Sensor leftTouch, ColorSensor color, UltrasonicSensor sonic)
- None move (self, distance)
- None turn (self, angle)
- None run (self)
- None stop (self)
- None process_behavior (self)
- None update_sensors (self)
- None update_queue (self)

Public Attributes

- **left_motor**
- **right_motor**
- **navigator**
- **frontTouch**
- **leftTouch**
- **color**
- **sonic**
- **queue**
- **hasHitFrontWall**
- **hasHitLeftWall**
- **distanceToWall**
- **current_color**
- **wallFollowingDistance**
- **isFollowingWall**
- **isWandering**
- **isFollowingFire**
- **fireNotFound**

5.3.1 Detailed Description

```
Custom defined Robot class for the lego ev3 robot. Responsible for moving and turning the robot.
```

5.3.2 Member Function Documentation

5.3.2.1 move()

```
None robotics.Robot.move (
              self,
              distance )
```

```
Moves robot a given distance in [mm]\n
Args: distance (float): Distance to be traveled [mm]
```

5.3.2.2 process_behavior()

```
None robotics.Robot.process_behavior (
              self )
```

```
Proccess a behavior from the priority queue. Pops the highest priority
behavior from queue then processes the behavior in RobotBehavior classes.
```

5.3.2.3 run()

```
None robotics.Robot.run (
              self )
```

```
The motor accelerates to TIRE_RPM
and keeps running at this speed
until you give a new command.
```

5.3.2.4 stop()

```
None robotics.Robot.stop (
              self )
```

```
Stops the motor and lets it spin freely.
The motor gradually stops due to friction.
```

5.3.2.5 turn()

```
None robotics.Robot.turn (
              self,
              angle )
```

```
Turns robot a given angle [deg]\n
Args: angle (float): angle to turn robot [deg]
```

5.3.2.6 update_queue()

```
None robotics.Robot.update_queue (
              self )
```

```
Updates the priority queue using the robot's sensor values. Defaults to Wander if
priority queue is empty.
```

5.3.2.7 update_sensors()

```
None robotics.Robot.update_sensors (
              self )
```

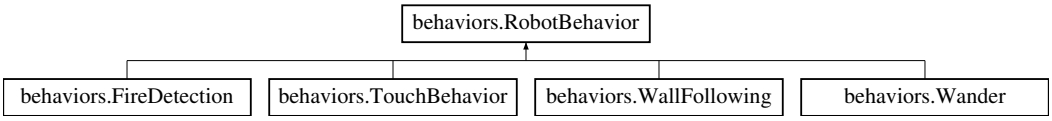
```
Function updates all of the robot's sensor values and stores these values in the robot object
```

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

- robotics.py

5.4 behaviors.RobotBehavior Class Reference

Inheritance diagram for behaviors.RobotBehavior:



Public Member Functions

- **__init__** (self, priority)
- run (self, robot)
- stop_behavior (self, robot)
- **__lt__** (self, other)

Public Attributes

- **priority**

5.4.1 Detailed Description

```
Coordinates the behaviors of the Robot given the priority of the behavior
```

5.4.2 Member Function Documentation

5.4.2.1 run()

```
behaviors.RobotBehavior.run (
              self,
              robot )

Starts a behavior and executes their corresponding functions\n
Args: robot (Robot): the robot object
```

Reimplemented in behaviors.TouchBehavior, behaviors.FireDetection, behaviors.WallFollowing, and behaviors.Wander.

5.4.2.2 stop_behavior()

```
behaviors.RobotBehavior.stop_behavior (
              self,
              robot )

Stops the current behavior and resets the neccessary variables\n
Args: robot (Robot): the robot object
```

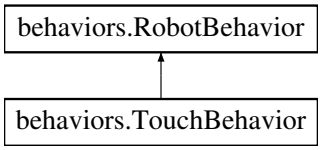
Reimplemented in behaviors.FireDetection, behaviors.WallFollowing, and behaviors.Wander.

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

- behaviors.py

5.5 behaviors.TouchBehavior Class Reference

Inheritance diagram for behaviors.TouchBehavior:



Public Member Functions

- __init__ (self)
- run (self, robot)
- None recalibrate_front (self, robot)
- None recalibrate_left (self, robot)

Public Member Functions inherited from behaviors.RobotBehavior

- stop_behavior (self, robot)
- **__lt__** (self, other)

Additional Inherited Members

Public Attributes inherited from behaviors.RobotBehavior

- **priority**

5.5.1 Detailed Description

```
Coordinates behaviors if robot touched a wall. Priority of 0
```

5.5.2 Constructor & Destructor Documentation

5.5.2.1 __init__()

```
behaviors.TouchBehavior.__init__ (
            self )
```

Reimplemented from behaviors.RobotBehavior.

5.5.3 Member Function Documentation

5.5.3.1 recalibrate_front()

```
None behaviors.TouchBehavior.recalibrate_front (
              self,
              robot )


Given that the robot ran into a wall,
backup the robot and turn it counterclockwise.\n
Args: robot (Robot): the robot object
```

5.5.3.2 recalibrate_left()

```
None behaviors.TouchBehavior.recalibrate_left (
              self,
              robot )


Given that the robot ran into a wall, turn the robot clockwise\n
Args: robot (Robot): the robot object
```

5.5.3.3 run()

```
behaviors.TouchBehavior.run (
              self,
              robot )


If recalibrate position and orientation of robot given if it hit a wall on the left or right\n
Args: robot (Robot): robot object
```

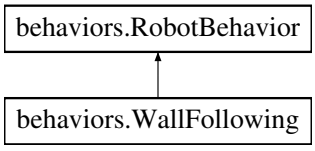
Reimplemented from behaviors.RobotBehavior.

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

- behaviors.py

5.6 behaviors.WallFollowing Class Reference

Inheritance diagram for behaviors.WallFollowing:



Public Member Functions

- __init__ (self)
- run (self, robot)
- stop_behavior (self, robot, msg)
- align_robot_with_wall (self, robot)
- None follow_wall (self, robot)

Public Member Functions inherited from behaviors.RobotBehavior

- **__lt__** (self, other)

Additional Inherited Members

Public Attributes inherited from behaviors.RobotBehavior

- **priority**

5.6.1 Detailed Description

```
Coordinates behaviors if robot gets close enough to the wall. Priority of 2.
```

5.6.2 Constructor & Destructor Documentation

5.6.2.1 __init__()

```
behaviors.WallFollowing.__init__ (
              self )
```

Reimplemented from behaviors.RobotBehavior.

5.6.3 Member Function Documentation

5.6.3.1 align_robot_with_wall()

```
behaviors.WallFollowing.align_robot_with_wall (
              self,
              robot )
```

```
Aligns the robot with the wall. Does arctan(Width / Distance) to find angle to turn\n
Args: robot (Robot): the robot object
```

5.6.3.2 follow_wall()

```
None behaviors.WallFollowing.follow_wall (
              self,
              robot )


Ensures the robot follows the wall until it detects that it should not
be following the wall anymore\n
Args: robot (Robot): the robot object
```

5.6.3.3 run()

```
behaviors.WallFollowing.run (
              self,
              robot )


Starts a behavior and executes their corresponding functions\n
Args: robot (Robot): the robot object
```

Reimplemented from behaviors.RobotBehavior.

5.6.3.4 stop_behavior()

```
behaviors.WallFollowing.stop_behavior (
              self,
              robot,
              msg )


Stops the current behavior and resets the neccessary variables\n
Args: robot (Robot): the robot object
```

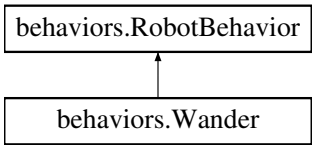
Reimplemented from behaviors.RobotBehavior.

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

- behaviors.py

5.7 behaviors.Wander Class Reference

Inheritance diagram for behaviors.Wander:



Public Member Functions

- __init__ (self)
- run (self, robot)
- wander (self, robot)
- stop_behavior (self, robot, msg)
- None recalibrate_front (self, robot)

Public Member Functions inherited from behaviors.RobotBehavior

- **__lt__** (self, other)

Public Attributes

- **start_time**
- **timeout_duration**

Public Attributes inherited from behaviors.RobotBehavior

- **priority**

5.7.1 Detailed Description

```
Default behavior of robot. Performs these actions until
the robot sensors detect somtheing. Priority of 3.
```

5.7.2 Constructor & Destructor Documentation

5.7.2.1 __init__()

```
behaviors.Wander.__init__ (
              self )
```

Reimplemented from behaviors.RobotBehavior.

5.7.3 Member Function Documentation

5.7.3.1 recalibrate_front()

```
None behaviors.Wander.recalibrate_front (
              self,
              robot )
```

```
Given that the robot ran into a wall,
backup the robot 200 mm and turn it 90 degrees.
```

5.7.3.2 run()

```
behaviors.Wander.run (
              self,
              robot )


Starts a behavior and executes their corresponding functions\n
Args: robot (Robot): the robot object
```

Reimplemented from behaviors.RobotBehavior.

5.7.3.3 stop_behavior()

```
behaviors.Wander.stop_behavior (
              self,
              robot,
              msg )


Stops the current behavior and resets the neccessary variables\n
Args: robot (Robot): the robot object
```

Reimplemented from behaviors.RobotBehavior.

5.7.3.4 wander()

```
behaviors.Wander.wander (
              self,
              robot )


Robot just moves forward until sensors
detect that a behavior should happen. Timeout the wandering behvior if the timeout time is reached.
Args: robot (Robot): the robot object
```

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

- behaviors.py

Index