ronelib

Generated by Doxygen 1.8.2

Thu Dec 6 2012 12:16:58

Contents

1 Main Page **1**

1.1 Overview: . . . 1

1.1.1 Software stack . . . 1

1.2 Included in ronelib: . . . 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 NbrNbr Struct Reference . . . 7

4.1.1 Detailed Description . . . 7

4.2 NbrNbrList Struct Reference . . . 7

4.2.1 Detailed Description . . . 7

5 File Documentation **9**

5.1 Behaviors/basicBehaviors.c File Reference . . . 9

5.1.1 Detailed Description . . . 9

5.2 Behaviors/basicBehaviors.h File Reference . . . 9

5.2.1 Detailed Description . . . 9

5.3 Behaviors/behaviorSystem.c File Reference . . . 9

5.3.1 Detailed Description . . . 10

5.4 Behaviors/behaviorSystem.h File Reference . . . 10

5.4.1 Detailed Description . . . 10

5.5 Behaviors/bumpBehaviors.c File Reference . . . 10

5.5.1 Detailed Description . . . 11

5.5.2 Function Documentation . . . 11

5.5.2.1 behBumpBackoff . . . 11

5.6 Behaviors/bumpBehaviors.h File Reference . . . 11

5.6.1 Detailed Description . . . 11

5.6.2 Function Documentation . . . 11

5.6.2.1 behBumpBackoff . . . 11
5.7 Behaviors/Navigation-midangle.c File Reference . . . 11
5.7.1 Detailed Description . . . 12
5.8 Behaviors/Navigation-midangle.h File Reference . . . 12
5.8.1 Detailed Description . . . 12
5.9 DataCollection/externalPose.c File Reference . . . 12
5.9.1 Detailed Description . . . 13
5.9.2 Function Documentation . . . 13
5.9.2.1 externalPoseGet . . . 13
5.9.2.2 externalPoseGetNbr . . . 13
5.9.2.3 externalPoseInit . . . 13
5.9.2.4 externalPoseIsActive . . . 13
5.9.2.5 externalPoseIsHost . . . 14
5.10 DataCollection/externalPose.h File Reference . . . 14
5.10.1 Detailed Description . . . 14
5.10.2 Function Documentation . . . 14
5.10.2.1 externalPoseGet . . . 14
5.10.2.2 externalPoseGetNbr . . . 15
5.10.2.3 externalPoseInit . . . 15
5.10.2.4 externalPoseIsActive . . . 15
5.10.2.5 externalPoseIsHost . . . 15
5.11 DataCollection/robotCanvas.c File Reference . . . 15
5.11.1 Detailed Description . . . 15
5.12 DataCollection/robotCanvas.h File Reference . . . 16
5.12.1 Detailed Description . . . 16
5.13 NeighborListOps/BroadcastComms.c File Reference . . . 16
5.13.1 Detailed Description . . . 16
5.13.2 Function Documentation . . . 17
5.13.2.1 broadcastMessageCreate . . . 17
5.14 NeighborListOps/BroadcastComms.h File Reference . . . 17
5.14.1 Detailed Description . . . 17
5.14.2 Function Documentation . . . 17
5.14.2.1 broadcastMessageCreate . . . 17
5.15 NeighborListOps/NbrNbrComms.c File Reference . . . 17
5.15.1 Detailed Description . . . 18
5.15.2 Function Documentation . . . 18
5.15.2.1 nbrNbrGetBearing . . . 18
5.15.2.2 nbrNbrGetID . . . 19
5.15.2.3 nbrNbrGetOrientation . . . 19
5.15.2.4 nbrNbrInit . . . 19

5.15.2.5 nbrNbrListGetNbrAtIdx . . . 19
5.15.2.6 nbrNbrListGetSize . . . 19
5.15.2.7 nbrNbrListPrint . . . 20
5.15.2.8 nbrNbrUpdate . . . 20
5.16 NeighborListOps/NbrNbrComms.h File Reference . . . 20
5.16.1 Detailed Description . . . 21
5.16.2 Typedef Documentation . . . 21
5.16.2.1 NbrNbr . . . 21
5.16.2.2 NbrNbrList . . . 21
5.16.3 Function Documentation . . . 21
5.16.3.1 nbrNbrGetBearing . . . 21
5.16.3.2 nbrNbrGetID . . . 22
5.16.3.3 nbrNbrGetOrientation . . . 22
5.16.3.4 nbrNbrInit . . . 22
5.16.3.5 nbrNbrListGetNbrAtIdx . . . 22
5.16.3.6 nbrNbrListGetSize . . . 22
5.16.3.7 nbrNbrListPrint . . . 23
5.17 NeighborListOps/neighborListOps.c File Reference . . . 23
5.17.1 Detailed Description . . . 24
5.17.2 Function Documentation . . . 24
5.17.2.1 nbrListAddNbr . . . 24
5.17.2.2 nbrListAverageBearing . . . 24
5.17.2.3 nbrListCopy . . . 24
5.17.2.4 nbrListGetClosestNbrToBearing . . . 25
5.17.2.5 nbrListGetFirst . . . 25
5.17.2.6 nbrListGetRobotWithID . . . 25
5.17.2.7 nbrListGetSecond . . . 25
5.17.2.8 nbrListGetSmallestAngleDeviation2 . . . 26
5.17.2.9 nbrListRemoveNbr . . . 26
5.18 NeighborListOps/neighborListOps.h File Reference . . . 26
5.18.1 Detailed Description . . . 27
5.18.2 Function Documentation . . . 27
5.18.2.1 nbrListAddNbr . . . 27
5.18.2.2 nbrListAverageBearing . . . 27
5.18.2.3 nbrListCopy . . . 27
5.18.2.4 nbrListGetClosestNbrToBearing . . . 28
5.18.2.5 nbrListGetFirst . . . 28
5.18.2.6 nbrListGetRobotWithID . . . 28
5.18.2.7 nbrListGetSecond . . . 28
5.18.2.8 nbrListGetSmallestAngleDeviation2 . . . 28

5.18.2.9 nbrListRemoveNbr 29

Index 29

Chapter 1

Main Page

1.1 Overview:

The r-one robots are designed by the Multi-Robotic Systems Lab at Rice University
http://mrsl.rice.edu/

1.1.1 Software stack

The code base is designed to be extensible, with a three-layer software stack:

- [Applications (i.e. `SensorTest`, `SuperDemo`)]
- [`ronelib` (basic behaviors that will be used to make other code)]
- [`roneos` (hardware, sensors, actuators, system-level code)]

1.2 Included in ronelib:

- `Behaviors`: for swarm functionality, moving, using the bump sensor to navigate, etc.
- `DataCollection`: radio commands and callbacks for external position
- `NeighborListOps`: functionality so robots can keep track of information on their neighbors in a local network

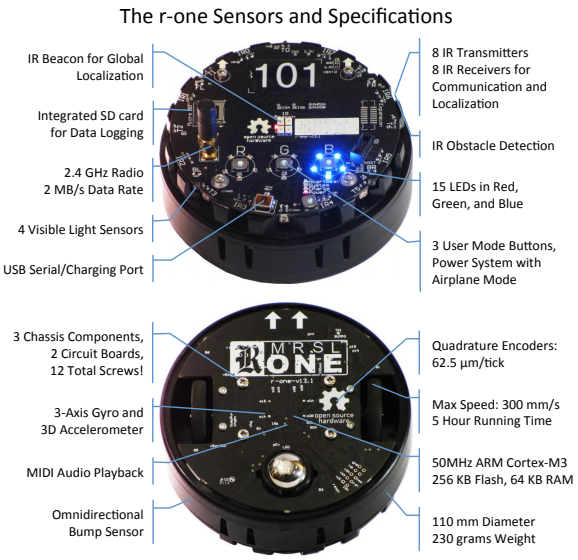

Figure 1.1: r-one robot specifications

Chapter 2

Class Index

2.1 Class List

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

NbrNbr
All relevant data about a neighbor's neighbor 7
NbrNbrList
Array of neighbor's neighbors 7

Chapter 3

File Index

3.1 File List

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

Behaviors/basicBehaviors.c
Basic swarm functionality behaviors: remote control, flocking, wall following, follow-the-leader . 9
Behaviors/basicBehaviors.h 9
Behaviors/behaviorSystem.c
Behaviors associated with moving 9
Behaviors/behaviorSystem.h 10
Behaviors/bumpBehaviors.c
Collection of behaviors that use the bump sensor (avoidance, navigation, etc.) 10
Behaviors/bumpBehaviors.h
Collection of behaviors that use the bump sensor (avoidance, navigation, etc.) 11
Behaviors/Navigation-midangle.c
Used to steer a robot through a network of robots without crashing 11
Behaviors/Navigation-midangle.h
Used to steer a robot through a network of robots without crashing 12
DataCollection/externalPose.c
Consistent API for updating and access the external pose of the robot swarm 12
DataCollection/externalPose.h 14
DataCollection/robotCanvas.c
Sends radio commands so that the robots can be plotted on a remote computer 15
DataCollection/robotCanvas.h
Sends radio commands so that the robots can be plotted on a remote computer 16
NeighborListOps/BroadcastComms.c
Used to communicate between robots 16
NeighborListOps/BroadcastComms.h 17
NeighborListOps/NbrNbrComms.c
The neighbor communication provides an API for accessing data about a robot's neighbors and information on those neighbor's neighbors 17
NeighborListOps/NbrNbrComms.h 20
NeighborListOps/neighborListOps.c
Operations on lists of neighbors: adding/removing, searching, looking for robots with certain bearings 23
NeighborListOps/neighborListOps.h 26

Chapter 4

Class Documentation

4.1 NbrNbr Struct Reference

all relevant data about a neighbor's neighbor

`#include <NbrNbrComms.h>`

Public Attributes

- int16 orientation

  TODO: what is the difference between bearing and orientation?

4.1.1 Detailed Description

all relevant data about a neighbor's neighbor

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

- NeighborListOps/NbrNbrComms.h

4.2 NbrNbrList Struct Reference

array of neighbor's neighbors

`#include <NbrNbrComms.h>`

4.2.1 Detailed Description

array of neighbor's neighbors

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

- NeighborListOps/NbrNbrComms.h

Chapter 5

File Documentation

5.1 Behaviors/basicBehaviors.c File Reference

basic swarm functionality behaviors: remote control, flocking, wall following, follow-the-leader

```
#include <stdio.h>
#include <stdlib.h>
#include "roneos.h"
#include "ronelib.h"
```

5.1.1 Detailed Description

basic swarm functionality behaviors: remote control, flocking, wall following, follow-the-leader this section is best for basic *swarm* behaviors. Other things that would fit here is disperse, cluster, sort

Since

Sep 28, 2011

Author

: jamesm

5.2 Behaviors/basicBehaviors.h File Reference

5.2.1 Detailed Description

Since

Sep 28, 2011

Author

jamesm

5.3 Behaviors/behaviorSystem.c File Reference

behaviors associated with moving

```
#include <stdio.h>
#include <stdlib.h>
#include "roneos.h"
#include "ronelib.h"
```

Functions

- Beh * behMoveForward (Beh *behPtr, int32 tv)
  Simple Behavior: move straight forward (no turning)

5.3.1 Detailed Description

behaviors associated with moving

Since

Sep 11, 2011

Author

jmclurkin

5.4 Behaviors/behaviorSystem.h File Reference

Functions

- Beh * behMoveForward (Beh *behPtr, int32 tv)
  Simple Behavior: move straight forward (no turning)

5.4.1 Detailed Description

Since

Sep 11, 2011

Author

jmclurkin

5.5 Behaviors/bumpBehaviors.c File Reference

collection of behaviors that use the bump sensor (avoidance, navigation, etc.)

```
#include <stdio.h>
#include <stdlib.h>
#include "roneos.h"
#include "ronelib.h"
```

Functions

- Beh * behBumpBackoff (Beh *behPtr, int32 tv)

5.5.1 Detailed Description

collection of behaviors that use the bump sensor (avoidance, navigation, etc.)

Since

Sep 11, 2011

Author

jamesm

5.5.2 Function Documentation

5.5.2.1 Beh* behBumpBackoff (Beh * *behPtr,* int32 *tv*)

added for improving bump behavior

5.6 Behaviors/bumpBehaviors.h File Reference

collection of behaviors that use the bump sensor (avoidance, navigation, etc.)

Functions

- Beh * behBumpBackoff (Beh *behPtr, int32 tv)

5.6.1 Detailed Description

collection of behaviors that use the bump sensor (avoidance, navigation, etc.)

Since

Sep 28, 2011

Author

jamesm

5.6.2 Function Documentation

5.6.2.1 Beh* behBumpBackoff (Beh * *behPtr,* int32 *tv*)

added for improving bump behavior

5.7 Behaviors/Navigation-midangle.c File Reference

used to steer a robot through a network of robots without crashing

```
#include <stdio.h>
#include <stdlib.h>
#include "roneos.h"
#include "ronelib.h"
```

5.7.1 Detailed Description

used to steer a robot through a network of robots without crashing

Since

Sep 22, 2011

Author

Golnaz Habibi

5.8 Behaviors/Navigation-midangle.h File Reference

used to steer a robot through a network of robots without crashing

5.8.1 Detailed Description

used to steer a robot through a network of robots without crashing

Since

Sep 23, 2011

Author

lyncas

5.9 DataCollection/externalPose.c File Reference

provides a consistent API for updating and access the external pose of the robot swarm.

```
#include <stdlib.h>
#include <ctype.h>
#include <string.h>
#include "roneos.h"
#include "ronelib.h"
```

Functions

- boolean externalPoseIsHost (void)

  determines if this is "the Host" (the robot connected over serial to external system)
- boolean externalPoseIsActive (void)

  is external post information active?
- void externalPoseInit (void)

  sets up memory and radio callbacks for getting external pose data. Should be called once in behaviorTask
- boolean externalPoseGet (ExternalPose *poseArg)

  gets this robot's pose
- boolean externalPoseGetNbr (ExternalPose *poseArg, Nbr *nbrPtr)

  gets this robot's pose

5.9.1 Detailed Description

provides a consistent API for updating and access the external pose of the robot swarm. This was developed as an interface between poses provided by the Newton Cam and the AprilTag system, but will work for any system that provides ASCII radio commands of the form: "EP, ID, X,Y,theta\n" Where 'X','Y','theta' are integers. To use this, call externalPoseInit(); in your behaviorTask() see "\robotcode\AprilTagReader\src\AprilTagReaderUsageExample.c"

Since

Nov 28, 2012

Author

jamesm

5.9.2 Function Documentation

5.9.2.1 boolean externalPoseGet (ExternalPose * *poseArg*)

gets this robot's pose

Parameters

poseArg	(overwritten if external pose is active with external pose data)

Returns

TRUE if external pose is active

5.9.2.2 boolean externalPoseGetNbr (ExternalPose * *poseArg,* Nbr * *nbrPtr*)

gets this robot's pose

Parameters

poseArg	(overwritten with external pose data if external pose is active and a neighbor)
nbrPtr,:	node we want to know the external pose of

Returns

TRUE if external pose is active AND this neighbor exists

5.9.2.3 void externalPoseInit (void)

sets up memory and radio callbacks for getting external pose data. Should be called once in behaviorTask

Returns

void

5.9.2.4 boolean externalPoseIsActive (void)

is external post information active?

Returns

TRUE if this is active or FALSE

5.9.2.5 boolean externalPoseIsHost (void)

determines if this is "the Host" (the robot connected over serial to external system)

Returns

TRUE if this is the host or FALSE

5.10 DataCollection/externalPose.h File Reference

Functions

- void externalPoseInit (void)
  sets up memory and radio callbacks for getting external pose data. Should be called once in behaviorTask
- boolean externalPoseGet (ExternalPose *poseArg)
  gets this robot's pose
- boolean externalPoseGetNbr (ExternalPose *poseArg, Nbr *nbrPtr)
  gets this robot's pose
- boolean externalPoseIsHost (void)
  determines if this is "the Host" (the robot connected over serial to external system)
- boolean externalPoseIsActive (void)
  is external post information active?

5.10.1 Detailed Description

Since

Nov 28, 2012

Author

jamesm

5.10.2 Function Documentation

5.10.2.1 boolean externalPoseGet (ExternalPose * *poseArg*)

gets this robot's pose

Parameters

poseArg	(overwritten if external pose is active with external pose data)

Returns

TRUE if external pose is active

5.10.2.2 boolean externalPoseGetNbr (ExternalPose * *poseArg,* Nbr * *nbrPtr*)

gets this robot's pose

Parameters

poseArg	(overwritten with external pose data if external pose is active and a neighbor)
nbrPtr,:	node we want to know the external pose of

Returns

TRUE if external pose is active AND this neighbor exists

5.10.2.3 void externalPoseInit (void)

sets up memory and radio callbacks for getting external pose data. Should be called once in behaviorTask

Returns

void

5.10.2.4 boolean externalPoseIsActive (void)

is external post information active?

Returns

TRUE if this is active or FALSE

5.10.2.5 boolean externalPoseIsHost (void)

determines if this is "the Host" (the robot connected over serial to external system)

Returns

TRUE if this is the host or FALSE

5.11 DataCollection/robotCanvas.c File Reference

sends radio commands so that the robots can be plotted on a remote computer

```
#include <stdio.h>
#include <stdlib.h>
#include "roneos.h"
#include "ronelib.h"
```

5.11.1 Detailed Description

sends radio commands so that the robots can be plotted on a remote computer

Since

Apr 23, 2011

Author

jamesm

5.12 DataCollection/robotCanvas.h File Reference

sends radio commands so that the robots can be plotted on a remote computer

5.12.1 Detailed Description

sends radio commands so that the robots can be plotted on a remote computer

Since

Apr 23, 2011

Author

jamesm

5.13 NeighborListOps/BroadcastComms.c File Reference

used to communicate between robots

```
#include <stdio.h>
#include <stdlib.h>
#include "roneos.h"
#include "ronelib.h"
```

Functions

- void broadcastMessageCreate (BroadcastMessage *msgPtr)

  creates and broadcasts the message in msgPtr

5.13.1 Detailed Description

used to communicate between robots

Since

Apr 6, 2011

Author

golnaz

5.13.2 Function Documentation

5.13.2.1 void broadcastMessageCreate (BroadcastMessage * *msgPtr*)

creates and broadcasts the message in msgPtr

Parameters

msgPtr	

Returns

void

5.14 NeighborListOps/BroadcastComms.h File Reference

```
#include "roneos.h"
```

Functions

- void broadcastMessageCreate (BroadcastMessage *msgPtr)
  creates and broadcasts the message in msgPtr

5.14.1 Detailed Description

Since

Apr 6, 2011

Author

jamesm

5.14.2 Function Documentation

5.14.2.1 void broadcastMessageCreate (BroadcastMessage * *msgPtr*)

creates and broadcasts the message in msgPtr

Parameters

msgPtr	

Returns

void

5.15 NeighborListOps/NbrNbrComms.c File Reference

The neighbor communication provides an API for accessing data about a robot's neighbors and information on those neighbor's neighbors.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "roneos.h"
#include "ronelib.h"
```

Functions

- void nbrNbrUpdate (NbrData *ndPtr)

  update this robot's nbrnbr data
- void nbrNbrInit (void)

  initializes messages for broadcasting and adds a callback
- uint8 nbrNbrListGetSize (NbrNbrList *nbrNbrListPtr)

  getter for the size of this neighbor's "neighbor list"
- void nbrNbrListPrint (NbrNbrList *nbrNbrListPtr)

  Print information on neighbor (and information of neighbor's neighbors).
- NbrNbr * nbrNbrListGetNbrAtIdx (NbrNbrList *nbrNbrListPtr, uint8 idx)

  returns pointer to the neighbors at index idx of this neighborList
- uint8 nbrNbrGetID (NbrNbr *nbrNbrPtr)

  Get ID of neighbor's neighbor.
- int16 nbrNbrGetBearing (NbrNbr *nbrNbrPtr)

  Get bearing of neighbor's neighbor.
- int16 nbrNbrGetOrientation (NbrNbr *nbrNbrPtr)

  Get orientation of neighbor's neighbor.

5.15.1 Detailed Description

The neighbor communication provides an API for accessing data about a robot's neighbors and information on those neighbor's neighbors. this code implements a class-type object and may be difficult to understand.

Since

Nov 13, 2012

Author

jamesm

5.15.2 Function Documentation

5.15.2.1 int16 nbrNbrGetBearing (NbrNbr * *nbrNbrPtr*)

Get bearing of neighbor's neighbor.

Warning

Incomplete.

Parameters

nbrNbrPtr	pointer to neighbor's neighbor

Returns

bearing

5.15.2.2 uint8 nbrNbrGetID (NbrNbr * *nbrNbrPtr*)

Get ID of neighbor's neighbor.

Parameters

nbrNbrPtr	pointer to neighbor's neighbor

Returns

the ID

5.15.2.3 int16 nbrNbrGetOrientation (NbrNbr * *nbrNbrPtr*)

Get orientation of neighbor's neighbor.

Parameters

nbrNbrPtr	pointer to neighbor's neighbor

Returns

orientation

5.15.2.4 void nbrNbrInit (void)

initializes messages for broadcasting and adds a callback

Returns

void

5.15.2.5 NbrNbr* nbrNbrListGetNbrAtIdx (NbrNbrList * *nbrNbrListPtr,* uint8 *idx*)

returns pointer to the neighbors at index idx of this neighborList

Parameters

nbrNbrListPtr	list we query
idx	index

Returns

pointer to the neighbors at index idx of this neighborList

5.15.2.6 uint8 nbrNbrListGetSize (NbrNbrList * *nbrNbrListPtr*)

getter for the size of this neighbor's "neighbor list"

Parameters

nbrNbrListPtr	list of the neighbors of this neighbor

Returns

number of neighbors, 0 if empty

5.15.2.7 void nbrNbrListPrint (NbrNbrList * *nbrNbrListPtr*)

Print information on neighbor (and information of neighbor's neighbors).

Print roneID and neighbor's ID, bear, orientation, orientation valid If neighbor_nbrnbr_enable is true, print ID and bearing of each neighbor's neighbor. Print name and value of each neighbor field.

Parameters

nbrNbrListPtr	neighbor pointer

Returns

void

5.15.2.8 void nbrNbrUpdate (NbrData * *ndPtr*)

update this robot's nbrnbr data

Parameters

ndPtr	pointer to neighbor's data

Returns

void

5.16 NeighborListOps/NbrNbrComms.h File Reference

Classes

- struct NbrNbr

  all relevant data about a neighbor's neighbor
- struct NbrNbrList

  array of neighbor's neighbors

Typedefs

- typedef struct NbrNbr NbrNbr

  all relevant data about a neighbor's neighbor
- typedef struct NbrNbrList NbrNbrList

  array of neighbor's neighbors

Functions

- void nbrNbrInit (void)

  initializes messages for broadcasting and adds a callback
- uint8 nbrNbrListGetSize (NbrNbrList *nbrNbrListPtr)

  getter for the size of this neighbor's "neighbor list"
- void nbrNbrListPrint (NbrNbrList *nbrNbrListPtr)

  Print information on neighbor (and information of neighbor's neighbors).
- NbrNbr * nbrNbrListGetNbrAtIdx (NbrNbrList *nbrNbrListPtr, uint8 idx)

  returns pointer to the neighbors at index idx of this neighborList
- uint8 nbrNbrGetID (NbrNbr *nbrNbrPtr)

  Get ID of neighbor's neighbor.
- int16 nbrNbrGetBearing (NbrNbr *nbrNbrPtr)

  Get bearing of neighbor's neighbor.
- int16 nbrNbrGetOrientation (NbrNbr *nbrNbrPtr)

  Get orientation of neighbor's neighbor.

5.16.1 Detailed Description

Since

Nov 13, 2012

Author

jamesm

5.16.2 Typedef Documentation

5.16.2.1 typedef struct NbrNbr NbrNbr

all relevant data about a neighbor's neighbor

5.16.2.2 typedef struct NbrNbrList NbrNbrList

array of neighbor's neighbors

5.16.3 Function Documentation

5.16.3.1 int16 nbrNbrGetBearing (NbrNbr * *nbrNbrPtr*)

Get bearing of neighbor's neighbor.

Warning

Incomplete.

Parameters

nbrNbrPtr	pointer to neighbor's neighbor

Returns

bearing

5.16.3.2 uint8 nbrNbrGetID (NbrNbr * *nbrNbrPtr*)

Get ID of neighbor's neighbor.

Parameters

nbrNbrPtr	pointer to neighbor's neighbor

Returns

the ID

5.16.3.3 int16 nbrNbrGetOrientation (NbrNbr * *nbrNbrPtr*)

Get orientation of neighbor's neighbor.

Parameters

nbrNbrPtr	pointer to neighbor's neighbor

Returns

orientation

5.16.3.4 void nbrNbrInit (void)

initializes messages for broadcasting and adds a callback

Returns

void

5.16.3.5 NbrNbr* nbrNbrListGetNbrAtIdx (NbrNbrList * *nbrNbrListPtr,* uint8 *idx*)

returns pointer to the neighbors at index idx of this neighborList

Parameters

nbrNbrListPtr	list we query
idx	index

Returns

pointer to the neighbors at index idx of this neighborList

5.16.3.6 uint8 nbrNbrListGetSize (NbrNbrList * *nbrNbrListPtr*)

getter for the size of this neighbor's "neighbor list"

Parameters

nbrNbrListPtr	list of the neighbors of this neighbor

Returns

number of neighbors, 0 if empty

5.16.3.7 void nbrNbrListPrint (NbrNbrList * *nbrNbrListPtr*)

Print information on neighbor (and information of neighbor's neighbors).

Print roneID and neighbor's ID, bear, orientation, orientation valid If neighbor_nbrnbr_enable is true, print ID and bearing of each neighbor's neighbor. Print name and value of each neighbor field.

Parameters

nbrNbrListPtr	neighbor pointer

Returns

void

5.17 NeighborListOps/neighborListOps.c File Reference

operations on lists of neighbors: adding/removing, searching, looking for robots with certain bearings.

```
#include <stdio.h>
#include <stdlib.h>
#include "roneos.h"
#include "ronelib.h"
```

Functions

- void nbrListAddNbr (NbrList *nbrListPtr, Nbr *nbrPtr)

  adds a neighbor to a neighbor's linked list of neighbors if there is room.
- void nbrListCopy (NbrList *nbrListDstPtr, NbrList *nbrListSrcPtr)

  copies the neighbor list from SRC to DST
- int16 nbrListAverageBearing (NbrList *nbrListPtr)

  determines average bearing of all neighbors using sum of vectors method (optimized).
- Nbr * nbrListGetSmallestAngleDeviation2 (NbrList *nbrListPtr, Nbr *nbrPtr)

  returns the neighbor whose bearing is closest to nbrPtr's bearing, NULL if list is empty.
- void nbrListRemoveNbr (NbrList *nbrListPtr, Nbr *nbrPtr)

  removes nbrPtr from list (if it exists)
- Nbr * nbrListGetClosestNbrToBearing (NbrList *nbrListPtr, int16 bearing)

  returns the neighbor whose bearing is closest to bearing, NULL if list is empty.
- Nbr * nbrListGetFirst (NbrList *nbrListPtr)

  returns a pointer to the neighbor at the head of the list, or NULL
- Nbr * nbrListGetSecond (NbrList *nbrListPtr)

  returns a pointer to the neighbor second in the list, or NULL
- Nbr * nbrListGetRobotWithID (NbrList *nbrListPtr, uint8 ID)

  returns a pointer to the neighbor with ID, or NULL

5.17.1 Detailed Description

operations on lists of neighbors: adding/removing, searching, looking for robots with certain bearings. ???

Since

Sep 11, 2011

Author

jmclurkin

5.17.2 Function Documentation

5.17.2.1 void nbrListAddNbr (NbrList * *nbrListPtr,* Nbr * *nbrPtr*)

adds a neighbor to a neighbor's linked list of neighbors if there is room.

Warning

: shouldn't we return 0 on success, and -1 on failure? I don't like voids (Aaron Becker)

Parameters

nbrListPtr	the neighbor's linked list
nbrPtr	a new neighbor

Returns

void

5.17.2.2 int16 nbrListAverageBearing (NbrList * *nbrListPtr*)

determines average bearing of all neighbors using sum of vectors method (optimized).

Parameters

nbrListPtr	list of neighbors

Returns

average bearing (milliradians)

5.17.2.3 void nbrListCopy (NbrList * *nbrListDstPtr,* NbrList * *nbrListSrcPtr*)

copies the neighbor list from SRC to DST

Parameters

nbrListSrcPtr	(unchanged)
nbrListDstPtr	becomes a copy of nbrListSrcPtr

Returns

void

5.17.2.4 Nbr* nbrListGetClosestNbrToBearing (NbrList * *nbrListPtr,* int16 *bearing*)

returns the neighbor whose bearing is closest to bearing, NULL if list is empty.

Parameters

nbrListPtr	list of neighbors
bearing	a reference bearing

Returns

pointer to neighbor whose bearing is closest to bearing, NULL if list is empty.

5.17.2.5 Nbr* nbrListGetFirst (NbrList * *nbrListPtr*)

returns a pointer to the neighbor at the head of the list, or NULL

Parameters

nbrListPtr	list of neighbors

Returns

a pointer to the neighbor at the head of the list, or NULL

5.17.2.6 Nbr* nbrListGetRobotWithID (NbrList * *nbrListPtr,* uint8 *ID*)

returns a pointer to the neighbor with ID, or NULL

Parameters

nbrListPtr	list of neighbors
ID	the unique number of the robot being searched for

Returns

a pointer to the neighbor with ID, or NULL

5.17.2.7 Nbr* nbrListGetSecond (NbrList * *nbrListPtr*)

returns a pointer to the neighbor second in the list, or NULL

Parameters

nbrListPtr	list of neighbors

Returns

a pointer to the neighbor second in the list, or NULL

5.17.2.8 Nbr* nbrListGetSmallestAngleDeviation2 (NbrList * *nbrListPtr,* Nbr * *nbrPtr*)

returns the neighbor whose bearing is closest to nbrPtr's bearing, NULL if list is empty.

Parameters

nbrListPtr	list of neighbors
nbrPtr	a neighbor used as a reference

Returns

pointer to neighbor whose bearing is closest to nbrPtr's bearing, NULL if list is empty.

5.17.2.9 void nbrListRemoveNbr (NbrList * *nbrListPtr,* Nbr * *nbrPtr*)

removes nbrPtr from list (if it exists)

Warning

: shouldn't we return 0 on success, and -1 on failure? I don't like voids (Aaron Becker)

Parameters

nbrListPtr	list of neighbors
nbrPtr	a neighbor to be removed

Returns

void

5.18 NeighborListOps/neighborListOps.h File Reference

Functions

- void nbrListAddNbr (NbrList *nbrListPtr, Nbr *nbrPtr)
  adds a neighbor to a neighbor's linked list of neighbors if there is room.
- void nbrListRemoveNbr (NbrList *nbrListPtr, Nbr *nbrPtr)
  removes nbrPtr from list (if it exists)
- void nbrListCopy (NbrList *nbrListDstPtr, NbrList *nbrListSrcPtr)
  copies the neighbor list from SRC to DST
- int16 nbrListAverageBearing (NbrList *nbrListPtr)
  determines average bearing of all neighbors using sum of vectors method (optimized).
- Nbr * nbrListGetFirst (NbrList *nbrListPtr)
  returns a pointer to the neighbor at the head of the list, or NULL
- Nbr * nbrListGetSecond (NbrList *nbrListPtr)
  returns a pointer to the neighbor second in the list, or NULL
- Nbr * nbrListGetRobotWithID (NbrList *nbrListPtr, uint8 ID)
  returns a pointer to the neighbor with ID, or NULL
- Nbr * nbrListGetSmallestAngleDeviation2 (NbrList *nbrListPtr, Nbr *nbrPtr)
  returns the neighbor whose bearing is closest to nbrPtr's bearing, NULL if list is empty.
- Nbr * nbrListGetClosestNbrToBearing (NbrList *nbrListPtr, int16 bearing)
  returns the neighbor whose bearing is closest to bearing, NULL if list is empty.

5.18.1 Detailed Description

Since

Sep 13, 2011

Author

jamesm

5.18.2 Function Documentation

5.18.2.1 void nbrListAddNbr (NbrList * *nbrListPtr,* Nbr * *nbrPtr*)

adds a neighbor to a neighbor's linked list of neighbors if there is room.

Warning

: shouldn't we return 0 on success, and -1 on failure? I don't like voids (Aaron Becker)

Parameters

nbrListPtr	the neighbor's linked list
nbrPtr	a new neighbor

Returns

void

5.18.2.2 int16 nbrListAverageBearing (NbrList * *nbrListPtr*)

determines average bearing of all neighbors using sum of vectors method (optimized).

Parameters

nbrListPtr	list of neighbors

Returns

average bearing (milliradians)

5.18.2.3 void nbrListCopy (NbrList * *nbrListDstPtr,* NbrList * *nbrListSrcPtr*)

copies the neighbor list from SRC to DST

Parameters

nbrListSrcPtr	(unchanged)
nbrListDstPtr	becomes a copy of nbrListSrcPtr

Returns

void

5.18.2.4 Nbr* nbrListGetClosestNbrToBearing (NbrList * *nbrListPtr,* int16 *bearing*)

returns the neighbor whose bearing is closest to bearing, NULL if list is empty.

Parameters

nbrListPtr	list of neighbors
bearing	a reference bearing

Returns

pointer to neighbor whose bearing is closest to bearing, NULL if list is empty.

5.18.2.5 Nbr* nbrListGetFirst (NbrList * *nbrListPtr*)

returns a pointer to the neighbor at the head of the list, or NULL

Parameters

nbrListPtr	list of neighbors

Returns

a pointer to the neighbor at the head of the list, or NULL

5.18.2.6 Nbr* nbrListGetRobotWithID (NbrList * *nbrListPtr,* uint8 *ID*)

returns a pointer to the neighbor with ID, or NULL

Parameters

nbrListPtr	list of neighbors
ID	the unique number of the robot being searched for

Returns

a pointer to the neighbor with ID, or NULL

5.18.2.7 Nbr* nbrListGetSecond (NbrList * *nbrListPtr*)

returns a pointer to the neighbor second in the list, or NULL

Parameters

nbrListPtr	list of neighbors

Returns

a pointer to the neighbor second in the list, or NULL

5.18.2.8 Nbr* nbrListGetSmallestAngleDeviation2 (NbrList * *nbrListPtr,* Nbr * *nbrPtr*)

returns the neighbor whose bearing is closest to nbrPtr's bearing, NULL if list is empty.

Parameters

nbrListPtr	list of neighbors
nbrPtr	a neighbor used as a reference

Returns

pointer to neighbor whose bearing is closest to nbrPtr's bearing, NULL if list is empty.

5.18.2.9 void nbrListRemoveNbr (NbrList * *nbrListPtr,* Nbr * *nbrPtr*)

removes nbrPtr from list (if it exists)

Warning

: shouldn't we return 0 on success, and -1 on failure? I don't like voids (Aaron Becker)

Parameters

nbrListPtr	list of neighbors
nbrPtr	a neighbor to be removed

Returns

void

Index

behBumpBackoff
    bumpBehaviors.c, 11
    bumpBehaviors.h, 11
Behaviors/Navigation-midangle.c, 11
Behaviors/Navigation-midangle.h, 12
Behaviors/basicBehaviors.c, 9
Behaviors/basicBehaviors.h, 9
Behaviors/behaviorSystem.c, 9
Behaviors/behaviorSystem.h, 10
Behaviors/bumpBehaviors.c, 10
Behaviors/bumpBehaviors.h, 11
BroadcastComms.c
    broadcastMessageCreate, 17
BroadcastComms.h
    broadcastMessageCreate, 17
broadcastMessageCreate
    BroadcastComms.c, 17
    BroadcastComms.h, 17
bumpBehaviors.c
    behBumpBackoff, 11
bumpBehaviors.h
    behBumpBackoff, 11

DataCollection/externalPose.c, 12
DataCollection/externalPose.h, 14
DataCollection/robotCanvas.c, 15
DataCollection/robotCanvas.h, 16

externalPose.c
    externalPoseGet, 13
    externalPoseGetNbr, 13
    externalPoseInit, 13
    externalPoseIsActive, 13
    externalPoseIsHost, 14
externalPose.h
    externalPoseGet, 14
    externalPoseGetNbr, 14
    externalPoseInit, 15
    externalPoseIsActive, 15
    externalPoseIsHost, 15
externalPoseGet
    externalPose.c, 13
    externalPose.h, 14
externalPoseGetNbr
    externalPose.c, 13
    externalPose.h, 14
externalPoseInit
    externalPose.c, 13
    externalPose.h, 15
externalPoseIsActive
    externalPose.c, 13
    externalPose.h, 15
externalPoseIsHost
    externalPose.c, 14
    externalPose.h, 15

nbrListAddNbr
    neighborListOps.c, 24
    neighborListOps.h, 27
nbrListAverageBearing
    neighborListOps.c, 24
    neighborListOps.h, 27
nbrListCopy
    neighborListOps.c, 24
    neighborListOps.h, 27
nbrListGetClosestNbrToBearing
    neighborListOps.c, 25
    neighborListOps.h, 27
nbrListGetFirst
    neighborListOps.c, 25
    neighborListOps.h, 28
nbrListGetRobotWithID
    neighborListOps.c, 25
    neighborListOps.h, 28
nbrListGetSecond
    neighborListOps.c, 25
    neighborListOps.h, 28
nbrListGetSmallestAngleDeviation2
    neighborListOps.c, 25
    neighborListOps.h, 28
nbrListRemoveNbr
    neighborListOps.c, 26
    neighborListOps.h, 29
NbrNbr, 7
    NbrNbrComms.h, 21
NbrNbrComms.c
    nbrNbrGetBearing, 18
    nbrNbrGetID, 19
    nbrNbrGetOrientation, 19
    nbrNbrInit, 19
    nbrNbrListGetNbrAtIdx, 19
    nbrNbrListGetSize, 19
    nbrNbrListPrint, 20
    nbrNbrUpdate, 20
NbrNbrComms.h
    NbrNbr, 21
    nbrNbrGetBearing, 21
    nbrNbrGetID, 22
    nbrNbrGetOrientation, 22
    nbrNbrInit, 22

NbrNbrList, 21
nbrNbrListGetNbrAtIdx, 22
nbrNbrListGetSize, 22
nbrNbrListPrint, 23
nbrNbrGetBearing
NbrNbrComms.c, 18
NbrNbrComms.h, 21
nbrNbrGetID
NbrNbrComms.c, 19
NbrNbrComms.h, 22
nbrNbrGetOrientation
NbrNbrComms.c, 19
NbrNbrComms.h, 22
nbrNbrInit
NbrNbrComms.c, 19
NbrNbrComms.h, 22
NbrNbrList, 7
NbrNbrComms.h, 21
nbrNbrListGetNbrAtIdx
NbrNbrComms.c, 19
NbrNbrComms.h, 22
nbrNbrListGetSize
NbrNbrComms.c, 19
NbrNbrComms.h, 22
nbrNbrListPrint
NbrNbrComms.c, 20
NbrNbrComms.h, 23
nbrNbrUpdate
NbrNbrComms.c, 20
neighborListOps.c
nbrListAddNbr, 24
nbrListAverageBearing, 24
nbrListCopy, 24
nbrListGetClosestNbrToBearing, 25
nbrListGetFirst, 25
nbrListGetRobotWithID, 25
nbrListGetSecond, 25
nbrListGetSmallestAngleDeviation2, 25
nbrListRemoveNbr, 26
neighborListOps.h
nbrListAddNbr, 27
nbrListAverageBearing, 27
nbrListCopy, 27
nbrListGetClosestNbrToBearing, 27
nbrListGetFirst, 28
nbrListGetRobotWithID, 28
nbrListGetSecond, 28
nbrListGetSmallestAngleDeviation2, 28
nbrListRemoveNbr, 29
NeighborListOps/BroadcastComms.c, 16
NeighborListOps/BroadcastComms.h, 17
NeighborListOps/NbrNbrComms.c, 17
NeighborListOps/NbrNbrComms.h, 20
NeighborListOps/neighborListOps.c, 23
NeighborListOps/neighborListOps.h, 26