Tennix Ball Assistant [[Group #4]] Kevin Albertson, Pawel Szczurko, Matthew Reyes, Sewon Chung, Edward Choi, Timothy Yong

Generated by Doxygen 1.8.6

Sun Mar 29 2015 06:12:49

Contents

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all 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

File Index

4.1 File List

Here is a list of all files with brief descriptions:

Chapter 5

Namespace Documentation

5.1 basket Namespace Reference

Functions

- def _basket_image_hue_filter

  Internal wrapper image hue filter.
- def _save_image

  Saves an image to the current directory.
- def _init_particle_filter

  Internal wrapper to particle filter initializer.
- def is_basket_middle

  Single entry function returning True/False if basket is in the middle of the screen.
- def run_middle

  Runs continuously and prints if the best detected blob is in the middle.
- def run

  Runs continuously outlines best matched blob if it is in the middle.

Variables

- particle_filter = None
- int image_half_size = -1
- int save_count = 1
- tuple base_filename = datetime.now()

5.1.1 Function Documentation

5.1.1.1 def basket._basket_image_hue_filter (*img*) `[private]`

Internal wrapper image hue filter.

param img SimpleCV.Image the image to apply the hue filter to

5.1.1.2 def basket._init_particle_filter (*img*) `[private]`

Internal wrapper to particle filter initializer.

Parameters

img	SimpleCV.Image Any image captured from the Camera, used to initialize the size

5.1.1.3 def basket._save_image (*img*) [private]

Saves an image to the current directory.

Parameters

img	SimpleCV.Image the image to save

5.1.1.4 def basket.is_basket_middle (*img*)

Single entry function returning True/False if basket is in the middle of the screen.

Parameters

img	SimpleCV.Image The image to test

5.1.1.5 def basket.run (*bestBlobCallback* = False)

Runs continuously outlines best matched blob if it is in the middle.

Parameters

bestBlobCallback	function Callback called passing the best blob found

5.1.1.6 def basket.run_middle ( )

Runs continuously and prints if the best detected blob is in the middle.

5.1.2 Variable Documentation

5.1.2.1 tuple basket.base_filename = datetime.now()

5.1.2.2 int basket.image_half_size = -1

5.1.2.3 basket.particle_filter = None

5.1.2.4 int basket.save_count = 1

5.2 basket_runner Namespace Reference

5.3 basket_test Namespace Reference

Functions

- def unitTest
- def basketPresent
- def basketMissing

5.3.1 Function Documentation

5.3.1.1 def basket_test.basketMissing ( )

5.3.1.2 def basket_test.basketPresent ( )

5.3.1.3 def basket_test.unitTest (*actual*, *expected*, *name*)

5.4 experiment Namespace Reference

Functions

- def experiment
- def hard_threshold
- def binary_mask
- def dilation_and_blur
- def blobs_by_mask

5.4.1 Detailed Description

```
A utility file for testing out computer vision techniques on preset images.
The purpose of this is to avoid using the webcam, and test on consistent test
cases.
```

5.4.2 Function Documentation

5.4.2.1 def experiment.binary_mask (*img*)

5.4.2.2 def experiment.blobs_by_mask (*img*)

5.4.2.3 def experiment.dilation_and_blur (*img*)

5.4.2.4 def experiment.experiment (*image_function* = None, *blob_function* = None, *directory* = "./")

5.4.2.5 def experiment.hard_threshold (*img*)

5.5 image_support Namespace Reference

Functions

- def external_init_particle_filter

  Initializes particle filter.
- def image_hue_filter

  Converts given image to HSV based on the given color.
- def get_hue_blobs

  Gets basket blobs after hue distance filtering.
- def get_best_blob

  Returns the best blob out of the provided set and particle filter.
- def is_blob_in_middle_helper

  Determines whether the given blob is in ceter of image.

5.5.1 Function Documentation

5.5.1.1 def image_support.external_init_particle_filter (*img*)

Initializes particle filter.

Parameters

img	SimpleCV.Image captured image

Returns

A ParticleFilter object

5.5.1.2 def image_support.get_best_blob (*blobs, particle_filter*)

Returns the best blob out of the provided set and particle filter.

Parameters

blobs	list of potential HSV blobs
particle_filter	initialized ParticleFilter object

Returns

The largest blob found or None.

5.5.1.3 def image_support.get_hue_blobs (*img*)

Gets basket blobs after hue distance filtering.

Parameters

img	SimpleCV.Image captured image.

Returns

Set of 'black' potential blobs.

5.5.1.4 def image_support.image_hue_filter (*img, ball* = `True`)

Converts given image to HSV based on the given color.

Parameters

img	SimpleCV.Image captured image
color	tuple of RGB values of singe 'H' value of HSV

Returns

HSV converted image

5.5.1.5 def image_support.is_blob_in_middle_helper (*img, blob*)

Determines whether the given blob is in ceter of image.

Parameters

img	SimpleCV.Image caputed image
blob	SimpleCV.Blob Blob object

Returns

True if blob in middle of image, false otherwise.

5.6 particlefilter Namespace Reference

Classes

- class ParticleFilter

5.7 prquadtree Namespace Reference

Classes

- class Point

  Represents an (x,y) coordinate point on a grid.
- class Particle

  Represents particle point.
- class Box

  Class defining a square on the coordinate system via a center point and half of square width.
- class PRQuadTree

  Class representing a Point Range Quadtree.

5.7.1 Detailed Description

```
Implementation of a Point Range Quadtree.

Author: Pawel Szczurko
```

5.8 prquadtree_test Namespace Reference

Classes

- class TestPoint
- class TestParticle
- class TestBox
- class TestPrQuadTree

5.9 prquadtree_test_example Namespace Reference

Variables

- tuple b = Box(Point(5,5), 50)
- tuple b2 = Box(Point(50,50), 50)
- tuple qt = PRQuadTree(b2)
- tuple pt = Point(2,2)
- tuple nearby = qt.query_k_nearest(pt, 20)
- int c = 1

5.9.1 Detailed Description

```
File testing the capabilities of the PRQuadTree.

Author: Pawel Szczurko
```

5.9.2 Variable Documentation

5.9.2.1 tuple prquadtree_test_example.b = Box(Point(5,5), 50)

5.9.2.2 tuple prquadtree_test_example.b2 = Box(Point(50,50), 50)

5.9.2.3 int prquadtree_test_example.c = 1

5.9.2.4 tuple prquadtree_test_example.nearby = qt.query_k_nearest(pt, 20)

5.9.2.5 tuple prquadtree_test_example.pt = Point(2,2)

5.9.2.6 tuple prquadtree_test_example.qt = PRQuadTree(b2)

5.10 tennis_ball Namespace Reference

Functions

- def _init_particle_filter

  Internal wrapper to particle filter initializer.
- def _ball_image_hue_filter

  Internal wrapper image hue filter.
- def is_ball_middle

  Entry point for module which determines whether tennis ball is in the middle of the image.
- def run

  Continuously captures image from computer camera and feeds it to the is_ball_middle method to detect whether tennis ball is in the middle of the screen.

Variables

- particle_filter = None

5.10.1 Detailed Description

```
Simple detection of ball using SimpleCV (much easier than OpenCV). The run method
identifies a tennis ball in the camera stream image. 'is_ball_middle' function
can be used to determine whether a ball is horizontally centered based on a specified
threshold.

-Pawel Szczurko
```

5.10.2 Function Documentation

5.10.2.1 def tennis_ball._ball_image_hue_filter (*img*) [private]

Internal wrapper image hue filter.

Parameters

img	SimpleCV.Image

Returns

img SimpleCV.Image converted to HSV

5.10.2.2 def tennis_ball._init_particle_filter (*img*) [private]

Internal wrapper to particle filter initializer.

Parameters

img	SimpleCV.Image

5.10.2.3 def tennis_ball.is_ball_middle (*img*)

Entry point for module which determines whether tennis ball is in the middle of the image.

Parameters

img	SimpleCV.Image

Returns

boolean. True if tennis ball is in middle, false otherwise.

5.10.2.4 def tennis_ball.run ( )

Continuously captures image from computer camera and feeds it to the is_ball_middle method to detect whether tennis ball is in the middle of the screen.

5.10.3 Variable Documentation

5.10.3.1 tennis_ball.particle_filter = None

5.11 tennis_ball_runner Namespace Reference

5.12 tennis_ball_test Namespace Reference

Functions

- def unitTest
- def ballPresent
- def ballMissing

5.12.1 Function Documentation

5.12.1.1 def tennis_ball_test.ballMissing ( )

5.12.1.2 def tennis_ball_test.ballPresent ( )

5.12.1.3 def tennis_ball_test.unitTest (*actual, expected, name*)

Chapter 6

Class Documentation

6.1 _ctrl Struct Reference

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

- SEVATBR-doc/agent/agent.c

6.2 prquadtree.Box Class Reference

Class defining a square on the coordinate system via a center point and half of square width.

Public Member Functions

- def __init__

  Construct a Box object.
- def contains_point

  Verifies that the given point is within this square.
- def intersect

  Checks if the provided box/square intersects with this square.

Public Attributes

- center
- half_size

6.2.1 Detailed Description

Class defining a square on the coordinate system via a center point and half of square width.

6.2.2 Constructor & Destructor Documentation

6.2.2.1 def prquadtree.Box.__init__ (*self*, *center*, *half_size*)

Construct a Box object.

Parameters

center	Point type specifying the center of the square
half_size	int half the length of the square

6.2.3 Member Function Documentation

6.2.3.1 def prquadtree.Box.contains_point (*self, point*)

Verifies that the given point is within this square.

Parameters

point	Point type to check if it's in the square

Returns

boolean indicating whether the point is within the square

6.2.3.2 def prquadtree.Box.intersect (*self, other_box*)

Checks if the provided box/square intersects with this square.

Parameters

other_box	Box object

Returns

Boolean indicating if the two intersect anywhere

6.2.4 Member Data Documentation

6.2.4.1 prquadtree.Box.center

6.2.4.2 prquadtree.Box.half_size

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

- SEVATBR-doc/visual/prquadtree.py

6.3 controller Struct Reference

```
#include <controller.h>
```

Public Attributes

- char ∗ name
- int32_t fd
- int8_t connected
- int32_t buttons
- int32_t axes
- int8_t A
- int8_t B

- int8_t X
- int8_t Y
- int8_t UP
- int8_t DOWN
- int8_t LEFT
- int8_t RIGHT
- int8_t LB
- int8_t RB
- float LT
- float RT
- struct {
  int8_t pressed
  float x
  float y
  } LJOY
- struct {
  int8_t pressed
  float x
  float y
  } RJOY
- int8_t START
- int8_t SELECT
- int8_t HOME
- int8_t LB2
- int8_t RB2

6.3.1 Member Data Documentation

6.3.1.1 int8_t controller::A

6.3.1.2 int32_t controller::axes

6.3.1.3 int8_t controller::B

6.3.1.4 int32_t controller::buttons

6.3.1.5 int8_t controller::connected

6.3.1.6 int8_t controller::DOWN

6.3.1.7 int32_t controller::fd

6.3.1.8 int8_t controller::HOME

6.3.1.9 int8_t controller::LB

6.3.1.10 int8_t controller::LB2

6.3.1.11 int8_t controller::LEFT

6.3.1.12 struct { ... } controller::LJOY

6.3.1.13 float controller::LT

6.3.1.14 char∗ controller::name

6.3.1.15 int8_t controller::pressed

6.3.1.16 int8_t controller::RB

6.3.1.17 int8_t controller::RB2

6.3.1.18 int8_t controller::RIGHT

6.3.1.19 struct { ... } controller::RJOY

6.3.1.20 float controller::RT

6.3.1.21 int8_t controller::SELECT

6.3.1.22 int8_t controller::START

6.3.1.23 int8_t controller::UP

6.3.1.24 int8_t controller::X

6.3.1.25 float controller::x

6.3.1.26 int8_t controller::Y

6.3.1.27 float controller::y

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

- SEVATBR-doc/manual/controller.h

6.4 HBridgeMotor Class Reference

Public Member Functions

- HBridgeMotor ()
- void setdigital (bool d)
- void write (int v)
- int attach (int pin1, int pin2)
- void reset ()

Public Attributes

- short velocity
- char pin [2]
- bool isdigital

6.4.1 Constructor & Destructor Documentation

6.4.1.1 HBridgeMotor::HBridgeMotor ( ) [inline]

6.4.2 Member Function Documentation

6.4.2.1 int HBridgeMotor::attach (int *pin1,* int *pin2*) [inline]

6.4.2.2 void HBridgeMotor::reset ( ) [inline]

6.4.2.3 void HBridgeMotor::setdigital (bool *d*) [inline]

6.4.2.4 void HBridgeMotor::write (int *v*) [inline]

6.4.3 Member Data Documentation

6.4.3.1 bool HBridgeMotor::isdigital

6.4.3.2 char HBridgeMotor::pin[2]

6.4.3.3 short HBridgeMotor::velocity

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

- SEVATBR-doc/robot/arduino/wheels/wheels.cpp

6.5 httplink Struct Reference

```
#include <httplink.h>
```

Public Attributes

- char ∗ hostname
- char ipaddr [128]
- int socket_fd
- int connected

6.5.1 Member Data Documentation

6.5.1.1 int httplink::connected

6.5.1.2 char∗ httplink::hostname

6.5.1.3 char httplink::ipaddr[128]

6.5.1.4 int httplink::socket_fd

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

- SEVATBR-doc/manual/httplink.h

6.6 info Struct Reference

Public Attributes

- speech_signal_t ss

6.6.1 Member Data Documentation

6.6.1.1 speech_signal_t info::ss

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

- SEVATBR-doc/agent/agent.c

6.7 note Union Reference

Public Attributes

- struct {
  char low
  char high
  };
- int16_t val

6.7.1 Member Data Documentation

6.7.1.1 struct { ... }

6.7.1.2 char note::high

6.7.1.3 char note::low

6.7.1.4 int16_t note::val

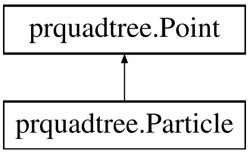
The documentation for this union was generated from the following file:

- SEVATBR-doc/speech/speech-to-text/lse_stt.c

6.8 prquadtree.Particle Class Reference

Represents particle point.

Inheritance diagram for prquadtree.Particle:



Public Member Functions

- def __init__

  Constructs a Particle.

Public Attributes

- x
- y
- score

6.8.1 Detailed Description

Represents particle point.

6.8.2 Constructor & Destructor Documentation

6.8.2.1 def prquadtree.Particle.__init__ (*self*, *x*, *y*)

Constructs a Particle.

Parameters

x	float/int x-position
y	float/int y-position

6.8.3 Member Data Documentation

6.8.3.1 prquadtree.Particle.score

6.8.3.2 prquadtree.Particle.x

6.8.3.3 prquadtree.Particle.y

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

- SEVATBR-doc/visual/prquadtree.py

6.9 particlefilter.ParticleFilter Class Reference

Public Member Functions

- def __init__

  Constructor.
- def iterate

  For each blob, it updates the points in the tree increasing the score of those which are within the bounding square of the blob.
- def score

  Returns the sum of the scores of the points found within this blob by querying the quadtree.
- def clear_scores

  Resets all scores of blobs This should be used when changing the webcam view.

Public Attributes

- pr_tree
- image_box
- iterations
- iterations_before_clearing

6.9.1 Constructor & Destructor Documentation

6.9.1.1 def particlefilter.ParticleFilter.__init__ (*self*, *box*)

Constructor.

Parameters

box	Box the box representing the web cam view

6.9.2 Member Function Documentation

6.9.2.1 def particlefilter.ParticleFilter.clear_scores (*self*)

Resets all scores of blobs This should be used when changing the webcam view.

6.9.2.2 def particlefilter.ParticleFilter.iterate (*self*, *blobs*)

For each blob, it updates the points in the tree increasing the score of those which are within the bounding square of the blob.

Parameters

blobs	array An array of blob objects which were matched

6.9.2.3 def particlefilter.ParticleFilter.score (*self*, *blob*)

Returns the sum of the scores of the points found within this blob by querying the quadtree.

Parameters

blob	Blob A single blob

Returns

int The sum of the scores of the points contained in the passed blob

6.9.3 Member Data Documentation

6.9.3.1 particlefilter.ParticleFilter.image_box

6.9.3.2 particlefilter.ParticleFilter.iterations

6.9.3.3 particlefilter.ParticleFilter.iterations_before_clearing

6.9.3.4 particlefilter.ParticleFilter.pr_tree

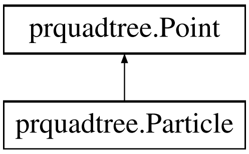
The documentation for this class was generated from the following file:

- SEVATBR-doc/visual/particlefilter.py

6.10 prquadtree.Point Class Reference

Represents an (x,y) coordinate point on a grid.

Inheritance diagram for prquadtree.Point:



Public Member Functions

- def __init__

  Constructs a coordinate Point.
- def __str__

  Overwritting the default to string method of the Point class.
- def __repr__

  Needed for printing via 'print'.

Public Attributes

- x
- y

6.10.1 Detailed Description

Represents an (x,y) coordinate point on a grid.

6.10.2 Constructor & Destructor Documentation

6.10.2.1 def prquadtree.Point.__init__ (*self*, *x*, *y*)

Constructs a coordinate Point.

Parameters

x	float/int x-position
y	float/int y-position

6.10.3 Member Function Documentation

6.10.3.1 def prquadtree.Point.__repr__ (*self*)

Needed for printing via 'print'.

6.10.3.2 def prquadtree.Point.__str__ (*self*)

Overwritting the default to string method of the Point class.

Returns

String representation of Point

6.10.4 Member Data Documentation

6.10.4.1 prquadtree.Point.x

6.10.4.2 prquadtree.Point.y

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

- SEVATBR-doc/visual/prquadtree.py

6.11 point2d Struct Reference

```
#include <coord.h>
```

Public Attributes

- double x
- double y

6.11.1 Member Data Documentation

6.11.1.1 double point2d::x

6.11.1.2 double point2d::y

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

- SEVATBR-doc/core/coord.h

6.12 point3d Struct Reference

```
#include <coord.h>
```

Public Attributes

- double x
- double y
- double z

6.12.1 Member Data Documentation

6.12.1.1 double point3d::x

6.12.1.2 double point3d::y

6.12.1.3 double point3d::z

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

- SEVATBR-doc/core/coord.h

6.13 pose2d Struct Reference

```
#include <coord.h>
```

Public Attributes

- double x
- double y
- double theta

6.13.1 Member Data Documentation

6.13.1.1 double pose2d::theta

6.13.1.2 double pose2d::x

6.13.1.3 double pose2d::y

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

- SEVATBR-doc/core/coord.h

6.14 pose3d Struct Reference

```
#include <coord.h>
```

Public Attributes

- double x
- double y
- double z
- double yaw
- double pitch
- double roll

6.14.1 Member Data Documentation

6.14.1.1 double pose3d::pitch

6.14.1.2 double pose3d::roll

6.14.1.3 double pose3d::x

6.14.1.4 double pose3d::y

6.14.1.5 double pose3d::yaw

6.14.1.6 double pose3d::z

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

- SEVATBR-doc/core/coord.h

6.15 prquadtree.PRQuadTree Class Reference

Class representing a Point Range Quadtree.

Public Member Functions

- def __init__

  Constructs a PR Quadtree given an initial square.
- def insert

  Inserts a point into the PRQuadtree.
- def query_range

  Returns the points in the provided range.
- def query_k_nearest

  Returns k points closest to the provided point.
- def print_all_points

  Prints all points stored in the PRQuadtree.
- def __str__

  Prints the points of the nw,ne,sw,se blocks of the given PRQuadTree node.

Static Public Member Functions

- def size

  Static method that determines the size of the given tree.

Public Attributes

- box
- points
- nw
- ne
- sw
- se

Static Public Attributes

- int QT_NODE_CAPACITY = 20

Private Member Functions

- def _subdivide

  Divides a node into nw,ne,sw,se pieces so that a new point can be inserted.

6.15.1 Detailed Description

Class representing a Point Range Quadtree.

6.15.2 Constructor & Destructor Documentation

6.15.2.1 def prquadtree.PRQuadTree.__init__ (*self, box*)

Constructs a PR Quadtree given an initial square.

Parameters

box	Box representing initial square

6.15.3 Member Function Documentation

6.15.3.1 def prquadtree.PRQuadTree.__str__ (*self*)

Prints the points of the nw,ne,sw,se blocks of the given PRQuadTree node.

Returns

String A string of points in the blocks Generates string based on the number of points stored in the provided node.

Parameters

loc	PRQuadTree a PRQuadTree node (ie nw,ne,sw,se)
name	String

Returns

String A string with point and name

6.15.3.2 def prquadtree.PRQuadTree._subdivide (*self*) [private]

Divides a node into nw,ne,sw,se pieces so that a new point can be inserted.

6.15.3.3 def prquadtree.PRQuadTree.insert (*self, point*)

Inserts a point into the PRQuadtree.

Parameters

point	Point

Returns

A boolean returning true on success, false on failure.

6.15.3.4 def prquadtree.PRQuadTree.print_all_points (*self, root*)

Prints all points stored in the PRQuadtree.

Parameters

root	PRQuadTree start point, or the root of the Quadtree

Returns

String a string with coordinates

6.15.3.5 def prquadtree.PRQuadTree.query_k_nearest (*self, point, k*)

Returns k points closest to the provided point.

Parameters

point	Point a Point from which to search for other points.
k	int number of closest points to return

Returns

array A coordinate distance between the search point and the provided point Internal method used to provide python method with a key (coordinate distance) on which to sort.

Parameters

p	Point

Returns

float

6.15.3.6 def prquadtree.PRQuadTree.query_range (*self, rng*)

Returns the points in the provided range.

Parameters

rng	Box a Box range from which to retrieve points

Returns

A list of points within the provided range

6.15.3.7 def prquadtree.PRQuadTree.size (*prtree*) [static]

Static method that determines the size of the given tree.

Keeping an insertion count in the client code would be preferred to this due to heavy recursion.

Parameters

prtree	PRQuadTree

Returns

int An integer representing the number of points in the given tree.

6.15.4 Member Data Documentation

6.15.4.1 prquadtree.PRQuadTree.box

6.15.4.2 prquadtree.PRQuadTree.ne

6.15.4.3 prquadtree.PRQuadTree.nw

6.15.4.4 prquadtree.PRQuadTree.points

6.15.4.5 int prquadtree.PRQuadTree.QT_NODE_CAPACITY = 20 [static]

6.15.4.6 prquadtree.PRQuadTree.se

6.15.4.7 prquadtree.PRQuadTree.sw

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

- SEVATBR-doc/visual/prquadtree.py

6.16 rawrec Struct Reference

```
#include <rawrec.h>
```

Public Attributes

- void ∗ pipeline
- void ∗ source
- void ∗ sink
- void ∗ buffer
- void ∗ caps
- char ∗ filesink_loc

6.16.1 Member Data Documentation

6.16.1.1 void∗ rawrec::buffer

6.16.1.2 void∗ rawrec::caps

6.16.1.3 char∗ rawrec::filesink_loc

6.16.1.4 void∗ rawrec::pipeline

6.16.1.5 void∗ rawrec::sink

6.16.1.6 void∗ rawrec::source

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

- SEVATBR-doc/speech/speech-to-text/rawrec.h

6.17 serial Struct Reference

```
#include <serial.h>
```

Public Attributes

- char ∗ port
- int fd
- int8_t connected
- int baudrate
- int parity
- char buffer [SWBUFMAX]
- char readbuf [SWREADMAX]
- int8_t readAvailable

6.17.1 Member Data Documentation

6.17.1.1 int serial::baudrate

6.17.1.2 char serial::buffer[SWBUFMAX]

6.17.1.3 int8_t serial::connected

6.17.1.4 int serial::fd

6.17.1.5 int serial::parity

6.17.1.6 char∗ serial::port

6.17.1.7 int8_t serial::readAvailable

6.17.1.8 char serial::readbuf[SWREADMAX]

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

- SEVATBR-doc/robot/serial/serial.h

6.18 speech_signal Struct Reference

```
#include <speech_signal.h>
```

Public Attributes

- unsigned char none
- unsigned char go
- unsigned char stop
- unsigned char fetch
- unsigned char ret

6.18.1 Member Data Documentation

6.18.1.1 unsigned char speech_signal::fetch

6.18.1.2 unsigned char speech_signal::go

6.18.1.3 unsigned char speech_signal::none

6.18.1.4 unsigned char speech_signal::ret

6.18.1.5 unsigned char speech_signal::stop

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

- SEVATBR-doc/speech/speech-to-text/speech_signal.h

6.19 stt Struct Reference

```
#include <lse_stt.h>
```

Public Attributes

- char nothing
- ps_decoder_t ∗ ps
- cmd_ln_t ∗ config

6.19.1 Member Data Documentation

6.19.1.1 cmd_ln_t∗ stt::config

6.19.1.2 char stt::nothing

6.19.1.3 ps_decoder_t∗ stt::ps

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

- SEVATBR-doc/speech/speech-to-text/lse_stt.h
- SEVATBR-doc/speech/speech-to-text/stt.h

6.20 tbr Struct Reference

```
#include <tbr.h>
```

Public Attributes

- serial_t ∗ connections
- int ∗ ids
- int8_t connected
- char ∗∗ possible_ports
- int num_possible
- int left
- int right
- int arm
- int claw
- int prev_left
- int prev_right
- int prev_arm
- int prev_claw

6.20.1 Member Data Documentation

6.20.1.1 int tbr::arm

6.20.1.2 int tbr::claw

6.20.1.3 int8_t tbr::connected

6.20.1.4 serial_t∗ tbr::connections

6.20.1.5 int∗ tbr::ids

6.20.1.6 int tbr::left

6.20.1.7 int tbr::num_possible

6.20.1.8 char∗∗ tbr::possible_ports

6.20.1.9 int tbr::prev_arm

6.20.1.10 int tbr::prev_claw

6.20.1.11 int tbr::prev_left

6.20.1.12 int tbr::prev_right

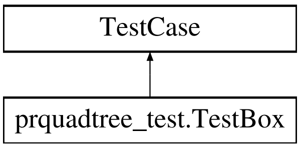
6.20.1.13 int tbr::right

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

- SEVATBR-doc/robot/tbr/tbr.h

6.21 prquadtree_test.TestBox Class Reference

Inheritance diagram for prquadtree_test.TestBox:



Public Member Functions

- def test_box_insert
- def test_box_contains

6.21.1 Member Function Documentation

6.21.1.1 def prquadtree_test.TestBox.test_box_contains (*self*)

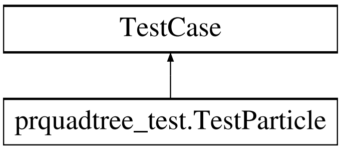
6.21.1.2 def prquadtree_test.TestBox.test_box_insert (*self*)

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

- SEVATBR-doc/visual/prquadtree_test.py

6.22 prquadtree_test.TestParticle Class Reference

Inheritance diagram for prquadtree_test.TestParticle:



Public Member Functions

- def test_particle_insert

6.22.1 Member Function Documentation

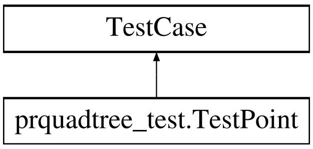
6.22.1.1 def prquadtree_test.TestParticle.test_particle_insert (*self*)

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

- SEVATBR-doc/visual/prquadtree_test.py

6.23 prquadtree_test.TestPoint Class Reference

Inheritance diagram for prquadtree_test.TestPoint:



Public Member Functions

- def test_point_insert

6.23.1 Member Function Documentation

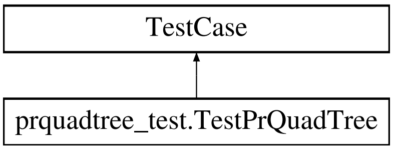
6.23.1.1 def prquadtree_test.TestPoint.test_point_insert (*self*)

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

- SEVATBR-doc/visual/prquadtree_test.py

6.24 prquadtree_test.TestPrQuadTree Class Reference

Inheritance diagram for prquadtree_test.TestPrQuadTree:



Public Member Functions

- def test_insert
- def test_nearby

6.24.1 Member Function Documentation

6.24.1.1 def prquadtree_test.TestPrQuadTree.test_insert (*self*)

6.24.1.2 def prquadtree_test.TestPrQuadTree.test_nearby (*self*)

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

- SEVATBR-doc/visual/prquadtree_test.py

Chapter 7

File Documentation

7.1 SEVATBR-doc/agent/agent.c File Reference

```
#include <pthread.h>
#include "speech.h"
#include "visual.h"
#include "coord.h"
#include "parietal.h"
```

Classes

- struct info
- struct _ctrl

Typedefs

- typedef struct info info_t

Functions

- void wakeup ()
- void ∗ concious_thought (void ∗information)
- void gotosleep ()

Variables

- static pthread_t conciousness
- static int end_conciousness
- struct _ctrl ctrl

7.1.1 Typedef Documentation

7.1.1.1 typedef struct info info_t

7.1.2 Function Documentation

7.1.2.1 void∗ concious_thought (void ∗ *information*)

Concious thought thread

Parameters

information	info for the thought

Returns

NULL

7.1.2.2 void gotosleep ( )

End threads, go back to sleep

7.1.2.3 void wakeup ( )

Wakeup from being asleep, start concious threads

7.1.3 Variable Documentation

7.1.3.1 pthread_t conciousness [static]

7.1.3.2 struct _ctrl ctrl

7.1.3.3 int end_conciousness [static]

7.2 SEVATBR-doc/agent/agent.h File Reference

```
#include "coord.h"
```

Macros

- #define AGENT_SIMPLE 0x0001

Functions

- int agent_create (int type)
- void agent_enable (void)
- void agent_disable (void)
- void agent_destroy (void)
- void agent_get_poses (pose3d_t ∗base, pose3d_t ∗arm)

7.2.1 Macro Definition Documentation

7.2.1.1 #define AGENT_SIMPLE 0x0001

7.2.2 Function Documentation

7.2.2.1 int agent_create (int *type*)

Create the agent

Parameters

type	the type to start the agent with

7.2.2.2 void agent_destroy (void)

Destroy the agent

7.2.2.3 void agent_disable (void)

Disable the agent

7.2.2.4 void agent_enable (void)

Enable the agent

7.2.2.5 void agent_get_poses (pose3d_t ∗ *base,* pose3d_t ∗ *arm*)

Get the poses from the agent

Parameters

base	the base pose to set
arm	the arm pose to set

7.3 SEVATBR-doc/agent/simple_agent.cpp File Reference

```
#include <sys/time.h>
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include "agent.h"
#include "tbd.h"
```

Functions

- int agent_create (int type)
- void agent_enable (void)
- void agent_disable (void)
- void agent_destroy (void)
- void agent_get_poses (pose3d_t ∗base, pose3d_t ∗arm)

Variables

- static int agent_enabled
- static int view [2]
- static struct timeval last_signal

7.3.1 Function Documentation

7.3.1.1 int agent_create (int *type*)

Create the agent

Parameters

type	the type to start the agent with

7.3.1.2 void agent_destroy (void)

Destroy the agent

7.3.1.3 void agent_disable (void)

Disable the agent

7.3.1.4 void agent_enable (void)

Enable the agent

7.3.1.5 void agent_get_poses (pose3d_t ∗ *base,* pose3d_t ∗ *arm*)

Get the poses from the agent

Parameters

base	the base pose to set
arm	the arm pose to set

7.3.2 Variable Documentation

7.3.2.1 int agent_enabled `[static]`

7.3.2.2 struct timeval last_signal `[static]`

7.3.2.3 int view[2] `[static]`

7.4 SEVATBR-doc/agent/test.c File Reference

```
#include <stdio.h>
#include <signal.h>
#include <sys/time.h>
#include "tbd.h"
```

Functions

- void stopprog (int signum)
- int main (int argc, char ∗argv[ ])

Variables

- static int exit_signal

7.4.1 Function Documentation

7.4.1.1 int main (int *argc*, char ∗ *argv[]*)

7.4.1.2 void stopprog (int *signum*)

7.4.2 Variable Documentation

7.4.2.1 int exit_signal [static]

7.5 SEVATBR-doc/manual/test.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <signal.h>
#include "manual.h"
```

Functions

- void stop (int signo)
- int main ()

Variables

- static int stopsig

7.5.1 Function Documentation

7.5.1.1 int main ( )

7.5.1.2 void stop (int *signo*)

7.5.2 Variable Documentation

7.5.2.1 int stopsig [static]

7.6 SEVATBR-doc/robot/tbr/test.c File Reference

```
#include "tbr.h"
```

Functions

- int main ()

7.6.1 Function Documentation

7.6.1.1 int main ( )

7.7 SEVATBR-doc/speech/speech-to-text/test.c File Reference

```
#include <stdio.h>
#include <signal.h>
#include "speech_signal.h"
```

Functions

- void stop (int param)
- int main (int argc, char ∗∗argv)

Variables

- static unsigned char stopsig

7.7.1 Function Documentation

7.7.1.1 int main (int *argc,* char ∗∗ *argv*)

7.7.1.2 void stop (int *param*)

7.7.2 Variable Documentation

7.7.2.1 unsigned char stopsig [static]

7.8 SEVATBR-doc/core/coord.h File Reference

Classes

- struct pose3d
- struct pose2d
- struct point3d
- struct point2d

Typedefs

- typedef struct pose3d pose3d_t
- typedef struct pose2d pose2d_t
- typedef struct point3d point3d_t
- typedef struct point2d point2d_t
- typedef point2d_t point_t
- typedef pose2d_t pose_t

7.8.1 Typedef Documentation

7.8.1.1 typedef struct point2d point2d_t

7.8.1.2 typedef struct point3d point3d_t

7.8.1.3 typedef point2d_t point_t

7.8.1.4 typedef struct pose2d pose2d_t

7.8.1.5 typedef struct pose3d pose3d_t

7.8.1.6 typedef pose2d_t pose_t

7.9 SEVATBR-doc/core/core.c File Reference

```
#include <signal.h>
#include <string.h>
#include <stdio.h>
#include "robot.h"
#include "manual.h"
#include "coord.h"
```

Functions

- void stop_program (int signum)
- int main (int argc, char ∗argv[])

Variables

- static int stop_signal

7.9.1 Function Documentation

7.9.1.1 int main (int *argc,* char ∗ *argv[]*)

This is the starting program for the robot

Parameters

argc	standard exec argument number
argv	standard exec argument list

Returns

0 on success, -1 otherwise

7.9.1.2 void stop_program (int *signum*)

Signal handler to stop the program

Parameters

signum	the signal number (kernel dependent)

7.9.2 Variable Documentation

7.9.2.1 int stop_signal [static]

7.10 SEVATBR-doc/core/core.cpp File Reference

```
#include <signal.h>
#include <string.h>
#include <stdio.h>
#include "robot.h"
#include "manual.h"
#include "coord.h"
```

Functions

- void stop_program (int signum)
- int main (int argc, char ∗argv[])

Variables

- static int stop_signal

7.10.1 Function Documentation

7.10.1.1 int main (int *argc,* char ∗ *argv[]*)

This is the starting program for the robot

Parameters

argc	standard exec argument number
argv	standard exec argument list

Returns

0 on success, -1 otherwise

7.10.1.2 void stop_program (int *signum*)

Signal handler to stop the program

Parameters

signum	the signal number (kernel dependent)

7.10.2 Variable Documentation

7.10.2.1 int stop_signal [static]

7.11 SEVATBR-doc/logger/logger.c File Reference

```
#include "logger.h"
```

Functions

- void logger_init (char ∗filename)
- void logger_print (char ∗msg)
- char ∗ logger_scan (void)
- void logger_destroy (void)

Variables

- static FILE ∗ logfile

7.11.1 Function Documentation

7.11.1.1 void logger_destroy (void)

Stop the logger

7.11.1.2 void logger_init (char ∗ *fname*)

Start the logger

Parameters

fname	name of the input file

7.11.1.3 void logger_print (char ∗ *msg*)

Print something to the logger

Parameters

msg	the message to print to the file

7.11.1.4 char ∗ logger_scan (void)

Get the current logger file

Returns

the entire message on success, NULL otherwise

7.11.2 Variable Documentation

7.11.2.1 FILE∗ logfile `[static]`

7.12 SEVATBR-doc/logger/logger.h File Reference

Functions

- void log_init (void)
- void log_print (char ∗msg)
- char ∗ log_scan (void)
- void log_destroy (void)

7.12.1 Function Documentation

7.12.1.1 void log_destroy (void)

7.12.1.2 void log_init (void)

7.12.1.3 void log_print (char ∗ *msg*)

7.12.1.4 char∗ log_scan (void)

7.13 SEVATBR-doc/manual/controller.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <dirent.h>
#include <linux/joystick.h>
#include <string.h>
#include <unistd.h>
#include <fcntl.h>
#include "controller.h"
```

Macros

- #define INPUT_DIR "/dev/input/"
- #define JS_PREFIX "js"
- #define MAX_16BIT 0x7FFF

Functions

- int controller_update (controller_t ∗ctrl)
- void controller_connect (controller_t ∗ctrl)
- void controller_disconnect (controller_t ∗ctrl)

7.13.1 Macro Definition Documentation

7.13.1.1 #define INPUT_DIR "/dev/input/"

7.13.1.2 #define JS_PREFIX "js"

7.13.1.3 #define MAX_16BIT 0x7FFF

7.13.2 Function Documentation

7.13.2.1 void controller_connect (controller_t ∗ *ctrl*)

Connect to a joystick device.

Parameters

controller	A pointer to a controller struct.

7.13.2.2 void controller_disconnect (controller_t ∗ *ctrl*)

Disconnect from a joystick device.

Parameters

controller	A pointer to a controller struct.

7.13.2.3 int controller_update (**controller_t** ∗ *ctrl*)

Hidden. Update a joystick device asynchronously.

Parameters

controller_arg	A (void ∗) pointer to a controller struct.

Returns

0 on success, else -1

7.14 SEVATBR-doc/manual/controller.h File Reference

```
#include <stdint.h>
#include <pthread.h>
```

Classes

- struct controller

Typedefs

- typedef struct controller controller_t

Functions

- void controller_connect (controller_t ∗ctrl)
- void controller_disconnect (controller_t ∗ctrl)

7.14.1 Typedef Documentation

7.14.1.1 typedef struct controller controller_t

7.14.2 Function Documentation

7.14.2.1 void controller_connect (**controller_t** ∗ *ctrl*)

Connect to a joystick device.

Parameters

controller	A pointer to a controller struct.

7.14.2.2 void controller_disconnect (**controller_t** ∗ *ctrl*)

Disconnect from a joystick device.

Parameters

controller	A pointer to a controller struct.

7.15 SEVATBR-doc/manual/httplink.c File Reference

```
#include <stdio.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <stdlib.h>
#include <netdb.h>
#include <string.h>
#include <unistd.h>
#include "httplink.h"
```

Functions

- int httplink_connect (httplink_t ∗connection, char ∗hostname)
- int httplink_send (httplink_t ∗connection, char ∗addr, char ∗type, char ∗data)
- char ∗ httplink_recv (httplink_t ∗connection)
- int httplink_disconnect (httplink_t ∗connection)

Variables

- static char msgbuf [1024]
- static char response [1024]

7.15.1 Function Documentation

7.15.1.1 int httplink_connect (httplink_t ∗ *connection,* char ∗ *hostname*)

Connect to the main server for our robot's manual interface.

Parameters

connection	the connection information for the server
hostname	the hostname of the server

Returns

0 on success, -1 otherwise

7.15.1.2 int httplink_disconnect (httplink_t ∗ *connection*)

Disconnect the connection

Parameters

connection	the connection information for the server

7.15.1.3 char∗ httplink_recv (httplink_t ∗ *connection*)

Try and receive a response from the main server

Parameters

connection	the connection information for the server

Returns

n bytes received, -1 otherwise

7.15.1.4 int httplink_send (**httplink_t** ∗ *connection,* char ∗ *addr,* char ∗ *type,* char ∗ *data*)

Send a request to the main server

Parameters

connection	the connection information for the server
addr	the addr to send the request to, "/" for main page
type	either "GET" or "POST" or "get" or "post"
data	the data to send over (only for httplink_POST)

Returns

n bytes sent over, -1 otherwise

7.15.2 Variable Documentation

7.15.2.1 char msgbuf[1024] `[static]`

7.15.2.2 char response[1024] `[static]`

7.16 SEVATBR-doc/manual/httplink.h File Reference

Classes

- struct httplink

Typedefs

- typedef struct httplink httplink_t

Functions

- int httplink_connect (httplink_t ∗connection, char ∗hostname)
- int httplink_send (httplink_t ∗connection, char ∗addr, char ∗type, char ∗data)
- char ∗ httplink_recv (httplink_t ∗connection)
- int httplink_disconnect (httplink_t ∗connection)

7.16.1 Typedef Documentation

7.16.1.1 typedef struct **httplink httplink_t**

7.16.2 Function Documentation

7.16.2.1 int httplink_connect (**httplink_t** ∗ *connection,* char ∗ *hostname*)

Connect to the main server for our robot's manual interface.

Parameters

connection	the connection information for the server
hostname	the hostname of the server

Returns

0 on success, -1 otherwise

7.16.2.2 int httplink_disconnect (**httplink_t** ∗ *connection*)

Disconnect the connection

Parameters

connection	the connection information for the server

7.16.2.3 char∗ httplink_recv (**httplink_t** ∗ *connection*)

Try and receive a response from the main server

Parameters

connection	the connection information for the server

Returns

n bytes received, -1 otherwise

7.16.2.4 int httplink_send (**httplink_t** ∗ *connection,* char ∗ *addr,* char ∗ *type,* char ∗ *data*)

Send a request to the main server

Parameters

connection	the connection information for the server
addr	the addr to send the request to, "/" for main page
type	either "GET" or "POST" or "get" or "post"
data	the data to send over (only for httplink_POST)

Returns

n bytes sent over, -1 otherwise

7.17 SEVATBR-doc/manual/manual.c File Reference

```
#include <stdio.h>
#include <signal.h>
#include <string.h>
#include <sys/time.h>
#include <stdlib.h>
#include "httplink.h"
#include "controller.h"
#include "manual.h"
```

Macros

- #define HZ 10

Functions

- static void server_update (void)
- static void raise_server_request (int signum)
- static void controller_update (void)
- int manual_connect (int id)
- void manual_enable (void)
- void manual_disable (void)
- int manual_disconnect (void)
- int manual_new_data (void)
- int isOverriden (void)
- void manual_get_poses (pose3d_t ∗b, pose3d_t ∗a)

Variables

- static int input_id
- static httplink_t server
- static controller_t ctrl
- static pose3d_t base
- static pose3d_t arm
- static int new_join
- static struct timeval last_signal
- static int manual_en
- static int mnl_override

7.17.1 Macro Definition Documentation

7.17.1.1 #define HZ 10

7.17.2 Function Documentation

7.17.2.1 void controller_update (void) [static]

Private to set the base and arm using information from the controller

7.17.2.2 int isOverriden (void)

User override

Returns

1 if overridden, else 0

7.17.2.3 int manual_connect (int *id*)

Connect to the manual connection - do not enable just yet.

Parameters

id	the id of the manual connection to connect to

Returns

0 on success, -1 otherwise

7.17.2.4 void manual_disable (void)

Disable manual mode

7.17.2.5 int manual_disconnect (void)

Disconnect from the manual connection

Returns

0, else -1 on error

7.17.2.6 void manual_enable (void)

Enable manual mode

7.17.2.7 void manual_get_poses (pose3d_t * *b*, pose3d_t * *a*)

Get the poses

Parameters

the	structs needed to hold the poses

7.17.2.8 int manual_new_data (void)

Get the status of the join

Returns

1 if a new join exists, else 0

7.17.2.9 static void raise_server_request (int *signum*) [static]

Private method to handle server requesting

Parameters

signum	the id for the signal

7.17.2.10 static void server_update (void) [static]

Private method to get the information sent over from the server and set the robot with this information

7.17.3 Variable Documentation

7.17.3.1 pose3d_t arm [static]

7.17.3.2 pose3d_t base [static]

7.17.3.3 controller_t ctrl [static]

7.17.3.4 int input_id [static]

7.17.3.5 struct timeval last_signal [static]

7.17.3.6 int manual_en [static]

7.17.3.7 int mnl_override [static]

7.17.3.8 int new_join [static]

7.17.3.9 httplink_t server [static]

7.18 SEVATBR-doc/manual/manual.h File Reference

```
#include "coord.h"
```

Macros

- #define MNL_SRVR 0x0001
- #define MNL_CTRL 0x0002

Functions

- int manual_connect (int type)
- void manual_enable (void)
- void manual_disable (void)
- int manual_disconnect (void)
- int manual_new_data (void)
- void manual_get_poses (pose3d_t ∗base, pose3d_t ∗arm)
- int isOverriden (void)

7.18.1 Macro Definition Documentation

7.18.1.1 #define MNL_CTRL 0x0002

7.18.1.2 #define MNL_SRVR 0x0001

7.18.2 Function Documentation

7.18.2.1 int isOverriden (void)

User override

Returns

1 if overridden, else 0

7.18.2.2 int manual_connect (int *id*)

Connect to the manual connection - do not enable just yet.

Parameters

id	the id of the manual connection to connect to

Returns

0 on success, -1 otherwise

7.18.2.3 void manual_disable (void)

Disable manual mode

7.18.2.4 int manual_disconnect (void)

Disconnect from the manual connection

Returns

0, else -1 on error

7.18.2.5 void manual_enable (void)

Enable manual mode

7.18.2.6 void manual_get_poses (pose3d_t ∗ *b,* pose3d_t ∗ *a*)

Get the poses

Parameters

the	structs needed to hold the poses

7.18.2.7 int manual_new_data (void)

Get the status of the join

Returns

1 if a new join exists, else 0

7.19 SEVATBR-doc/manual/rtsp/rtsplink.c File Reference

```
#include <gst/gst.h>
```

7.20 SEVATBR-doc/robot/arduino/arm/arm.cpp File Reference

```
#include <Servo.h>
#include <string.h>
```

Macros

- #define DEV_ID 2
- #define ARM_R1 3
- #define ARM_L1 5
- #define ARM_R2 6
- #define ARM_L2 9
- #define ARM_R3 10
- #define ARM_L3 11

Functions

- int limit (int x, int a, int b)
- void setarm (int vel)
- void stoparm ()
- void raisearm ()
- void lowerarm ()
- void setup ()
- void loop ()

Variables

- Servo arm_l [3]
- Servo arm_r [3]
- unsigned long time
- char msg [64]

7.20.1 Macro Definition Documentation

7.20.1.1 #define ARM_L1 5

7.20.1.2 #define ARM_L2 9

7.20.1.3 #define ARM_L3 11

7.20.1.4 #define ARM_R1 3

7.20.1.5 #define ARM_R2 6

7.20.1.6 #define ARM_R3 10

7.20.1.7 #define DEV_ID 2

7.20.2 Function Documentation

7.20.2.1 int limit (int *x*, int *a*, int *b*)

7.20.2.2 void loop ( )

7.20.2.3 void lowerarm ( )

7.20.2.4 void raisearm ( )

7.20.2.5 void setarm (int *vel*)

7.20.2.6 void setup ( )

7.20.2.7 void stoparm ( )

7.20.3 Variable Documentation

7.20.3.1 Servo arm_l[3]

7.20.3.2 Servo arm_r[3]

7.20.3.3 char msg[64]

7.20.3.4 unsigned long time

7.21 SEVATBR-doc/robot/arduino/claw/claw.cpp File Reference

```
#include <Servo.h>
#include <string.h>
```

Macros

- #define DEV_ID 3
- #define CLAW_L 10
- #define BOT_L 2
- #define TOP_L 3
- #define CLAW_R 9
- #define BOT_R 0
- #define TOP_R 1

Functions

- int limit (int x, int a, int b)
- void setclaw (int vel)
- void stopclaw ()
- void openclaw ()
- void closeclaw ()
- void setup ()
- void loop ()

Variables

- Servo claw_l
- Servo claw_r
- unsigned long time
- char msg [64]

7.21.1 Macro Definition Documentation

7.21.1.1 #define BOT_L 2

7.21.1.2 #define BOT_R 0

7.21.1.3 #define CLAW_L 10

7.21.1.4 #define CLAW_R 9

7.21.1.5 #define DEV_ID 3

7.21.1.6 #define TOP_L 3

7.21.1.7 #define TOP_R 1

7.21.2 Function Documentation

7.21.2.1 void closeclaw ( )

7.21.2.2 int limit (int *x*, int *a*, int *b*)

7.21.2.3 void loop ( )

7.21.2.4 void openclaw ( )

7.21.2.5 void setclaw (int *vel*)

7.21.2.6 void setup ( )

7.21.2.7 void stopclaw ( )

7.21.3 Variable Documentation

7.21.3.1 Servo claw_l

7.21.3.2 Servo claw_r

7.21.3.3 char msg[64]

7.21.3.4 unsigned long time

7.22 SEVATBR-doc/robot/arduino/wheels/wheels.cpp File Reference

```
#include <string.h>
```

Classes

- class HBridgeMotor

Macros

- #define DEV_ID 1

- #define WHEEL_RT1 11
- #define WHEEL_RT2 10
- #define WHEEL_RM1 9
- #define WHEEL_RM2 8
- #define WHEEL_RB1 7
- #define WHEEL_RB2 6
- #define WHEEL_LT1 A0
- #define WHEEL_LT2 A1
- #define WHEEL_LM1 A2
- #define WHEEL_LM2 A3
- #define WHEEL_LB1 A4
- #define WHEEL_LB2 A5

Functions

- int limit (int x, int a, int b)
- void setwheels (int left, int right)
- void stopwheels ()
- void turnleft ()
- void turnright ()
- void forward ()
- void backward ()
- void setup ()
- void loop ()

Variables

- HBridgeMotor wheel_l [3]
- HBridgeMotor wheel_r [3]
- unsigned long time
- char msg [64]

7.22.1 Macro Definition Documentation

7.22.1.1 #define DEV_ID 1

7.22.1.2 #define WHEEL_LB1 A4

7.22.1.3 #define WHEEL_LB2 A5

7.22.1.4 #define WHEEL_LM1 A2

7.22.1.5 #define WHEEL_LM2 A3

7.22.1.6 #define WHEEL_LT1 A0

7.22.1.7 #define WHEEL_LT2 A1

7.22.1.8 #define WHEEL_RB1 7

7.22.1.9 #define WHEEL_RB2 6

7.22.1.10 #define WHEEL_RM1 9

7.22.1.11 #define WHEEL_RM2 8

7.22.1.12 #define WHEEL_RT1 11

7.22.1.13 #define WHEEL_RT2 10

7.22.2 Function Documentation

7.22.2.1 void backward ( )

7.22.2.2 void forward ( )

7.22.2.3 int limit (int *x*, int *a*, int *b*)

7.22.2.4 void loop ( )

7.22.2.5 void setup ( )

7.22.2.6 void setwheels (int *left*, int *right*)

7.22.2.7 void stopwheels ( )

7.22.2.8 void turnleft ( )

7.22.2.9 void turnright ( )

7.22.3 Variable Documentation

7.22.3.1 char msg[64]

7.22.3.2 unsigned long time

7.22.3.3 HBridgeMotor wheel_l[3]

7.22.3.4 HBridgeMotor wheel_r[3]

7.23 SEVATBR-doc/robot/robot.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "robot.h"
#include "tbr.h"
```

Functions

- int robot_set (uint32_t robotid)
- int robot_unset (void)
- int robot_move (pose3d_t ∗base, pose3d_t ∗arm)

Variables

- static void ∗ robot

- static uint32_t currid

7.23.1 Function Documentation

7.23.1.1 int robot_move (pose3d_t ∗ *base,* pose3d_t ∗ *arm*)

Move the robot by some velocity or to some position

Parameters

base	either a direction or velocity (base)
arm	either a direction or velocity (arm)

Returns

0 on success, -1 otherwise

7.23.1.2 int robot_set (uint32_t *robotid*)

Initialize the robot to some id which specifies the device

Parameters

robotid	the id of the device

Returns

0 on success, -1 otherwise

7.23.1.3 int robot_unset (void)

Remove the robot from the known space

Returns

0 on success, -1 otherwise

7.23.2 Variable Documentation

7.23.2.1 uint32_t currid [static]

7.23.2.2 void∗ robot [static]

7.24 SEVATBR-doc/robot/robot.h File Reference

```
#include <stdint.h>
#include "coord.h"
```

Macros

- #define NO_ROBOT 0x00000000
- #define STANDARD_OUT 0x00000001
- #define TENNIS_BALL_ROBOT 0x00000002
- #define TACHIKOMA 0x00000003

Functions

- int robot_set (uint32_t robotid)
- int robot_unset (void)
- int robot_move (pose3d_t ∗base, pose3d_t ∗arm)

7.24.1 Macro Definition Documentation

7.24.1.1 #define NO_ROBOT 0x00000000

7.24.1.2 #define STANDARD_OUT 0x00000001

7.24.1.3 #define TACHIKOMA 0x00000003

7.24.1.4 #define TENNIS_BALL_ROBOT 0x00000002

7.24.2 Function Documentation

7.24.2.1 int robot_move (pose3d_t ∗ *base,* pose3d_t ∗ *arm*)

Move the robot by some velocity or to some position

Parameters

base	either a direction or velocity (base)
arm	either a direction or velocity (arm)

Returns

0 on success, -1 otherwise

7.24.2.2 int robot_set (uint32_t *robotid*)

Initialize the robot to some id which specifies the device

Parameters

robotid	the id of the device

Returns

0 on success, -1 otherwise

7.24.2.3 int robot_unset (void)

Remove the robot from the known space

Returns

0 on success, -1 otherwise

7.25 SEVATBR-doc/robot/serial/serial.c File Reference

```
#include <termios.h>
#include <sys/ioctl.h>
#include <sys/types.h>
#include <sys/wait.h>
#include <unistd.h>
#include <stdlib.h>
#include <string.h>
#include <fcntl.h>
#include <dirent.h>
#include <stdio.h>
#include "serial.h"
```

Macros

- #define INPUT_DIR "/dev/"

Functions

- static int _serial_setattr (serial_t ∗connection)
- static void _serial_update (serial_t ∗connection)
- int serial_connect (serial_t ∗connection, char ∗port, int baudrate)
- char ∗ serial_read (serial_t ∗connection)
- void serial_write (serial_t ∗connection, char ∗message)
- void serial_disconnect (serial_t ∗connection)

Variables

- static char const ∗ PREFIXES [3]
- static char tempbuf [SWREADMAX]

7.25.1 Macro Definition Documentation

7.25.1.1 #define INPUT_DIR "/dev/"

7.25.2 Function Documentation

7.25.2.1 static int _serial_setattr (serial_t ∗ *connection*) [static]

Helper method to set the attributes of a serial connection, particularly for the arduino or similar device.

Parameters

connection	the serial port to connect to

Returns

0 on success, -1 on failure

7.25.2.2 static void _serial_update (serial_t ∗ *connection*) [static]

Method to update the readbuf of the serial communication, as well as the connection itself.

Parameters

connection	the serial struct

Note

the packets will be read in the following format: data
however, the
will be cut off

7.25.2.3 int serial_connect (serial_t * *connection,* char * *port,* int *baudrate*)

Connect to a serial device.

Parameters

connection	a pointer to the serial struct
port	a portname; if NULL, will open a random port
baudrate	the bits per second of information to transmit/receive

Returns

0 on success, -1 on failure

7.25.2.4 void serial_disconnect (serial_t * *connection*)

Disconnect from the USB Serial port.

Parameters

connection	A pointer to the serial struct.

7.25.2.5 char* serial_read (serial_t * *connection*)

Read a string from the serial communication link.

Parameters

connection	the serial connection to read a message from

Returns

the readbuf if a message exists, else NULL

7.25.2.6 void serial_write (serial_t * *connection,* char * *message*)

Write a message to the serial communication link.

Parameters

connection	the serial communication link to write to
message	the message to send over to the other side

Note

be sure the message has a '
' chararacter

7.25.3 Variable Documentation

7.25.3.1 char const∗ PREFIXES[3] [static]

Initial value:

```
= {
  "ttyACM",

  NULL
}
```

7.25.3.2 char tempbuf[SWREADMAX] [static]

7.26 SEVATBR-doc/robot/serial/serial.h File Reference

`#include <stdint.h>`

Classes

- struct serial

Macros

- #define SWBUFMAX 64
- #define SWREADMAX 32
- #define SWWRITEMAX 32

Typedefs

- typedef struct serial serial_t

Functions

- int serial_connect (serial_t ∗connection, char ∗port, int baudrate)
- char ∗ serial_read (serial_t ∗connection)
- void serial_write (serial_t ∗connection, char ∗message)
- void serial_disconnect (serial_t ∗connection)

7.26.1 Macro Definition Documentation

7.26.1.1 #define SWBUFMAX 64

7.26.1.2 #define SWREADMAX 32

7.26.1.3 #define SWWRITEMAX 32

7.26.2 Typedef Documentation

7.26.2.1 typedef struct serial serial_t

7.26.3 Function Documentation

7.26.3.1 int serial_connect (serial_t ∗ *connection,* char ∗ *port,* int *baudrate*)

Connect to a serial device.

Parameters

connection	a pointer to the serial struct
port	a portname; if NULL, will open a random port
baudrate	the bits per second of information to transmit/receive

Returns

0 on success, -1 on failure

7.26.3.2 void serial_disconnect (serial_t ∗ *connection*)

Disconnect from the USB Serial port.

Parameters

connection	A pointer to the serial struct.

7.26.3.3 char∗ serial_read (serial_t ∗ *connection*)

Read a string from the serial communication link.

Parameters

connection	the serial connection to read a message from

Returns

the readbuf if a message exists, else NULL

7.26.3.4 void serial_write (serial_t ∗ *connection,* char ∗ *message*)

Write a message to the serial communication link.

Parameters

connection	the serial communication link to write to
message	the message to send over to the other side

Note

be sure the message has a '
' chararacter

7.27 SEVATBR-doc/robot/tbr/tbr.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <sys/types.h>
#include <dirent.h>
#include <termios.h>
#include <time.h>
#include "tbr.h"
```

Macros

- #define NUM_DEV 3
- #define DEV_BAUD B38400
- #define WHEEL_DEVID 1
- #define ARM_DEVID 2
- #define CLAW_DEVID 3
- #define SYNC_NSEC 500000000

Functions

- int tbr_connect (tbr_t ∗robot)
- void tbr_send (tbr_t ∗robot)
- void tbr_recv (tbr_t ∗robot)
- void tbr_disconnect (tbr_t ∗robot)
- void tbr_reset (tbr_t ∗robot)

7.27.1 Macro Definition Documentation

7.27.1.1 #define ARM_DEVID 2

7.27.1.2 #define CLAW_DEVID 3

7.27.1.3 #define DEV_BAUD B38400

7.27.1.4 #define NUM_DEV 3

7.27.1.5 #define SYNC_NSEC 500000000

7.27.1.6 #define WHEEL_DEVID 1

7.27.2 Function Documentation

7.27.2.1 int tbr_connect (tbr_t ∗ *robot*)

Initialize the communication layer

Parameters

robot	the robot information

Returns

number of devices initialized, -1 on error

7.27.2.2 void tbr_disconnect (tbr_t ∗ *robot*)

Disconnect everything

Parameters

robot	the robot information

7.27.2.3 void tbr_recv (tbr_t ∗ *robot*)

Receive input from the communication layer

Parameters

robot	the robot information

7.27.2.4 void tbr_reset (tbr_t ∗ *robot*)

Reset the robot values

Parameters

robot	the robot information

7.27.2.5 void tbr_send (tbr_t ∗ *robot*)

Send output to the communication layer

Parameters

robot	the robot information

7.28 SEVATBR-doc/robot/tbr/tbr.h File Reference

```
#include <stdint.h>
#include "serial.h"
```

Classes

- struct tbr

Typedefs

- typedef struct tbr tbr_t

Functions

- int tbr_connect (tbr_t ∗robot)
- void tbr_send (tbr_t ∗robot)
- void tbr_recv (tbr_t ∗robot)
- void tbr_disconnect (tbr_t ∗robot)
- void tbr_reset (tbr_t ∗robot)

7.28.1 Typedef Documentation

7.28.1.1 typedef struct tbr tbr_t

7.28.2 Function Documentation

7.28.2.1 int tbr_connect (tbr_t ∗ *robot*)

Initialize the communication layer

Parameters

robot	the robot information

Returns

number of devices initialized, -1 on error

7.28.2.2 void tbr_disconnect (tbr_t ∗ *robot*)

Disconnect everything

Parameters

robot	the robot information

7.28.2.3 void tbr_recv (tbr_t ∗ *robot*)

Receive input from the communication layer

Parameters

robot	the robot information

7.28.2.4 void tbr_reset (tbr_t ∗ *robot*)

Reset the robot values

Parameters

robot	the robot information

7.28.2.5 void tbr_send (tbr_t ∗ *robot*)

Send output to the communication layer

Parameters

robot	the robot information

7.29 SEVATBR-doc/slam/slam.c File Reference

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

Functions

- void set_noise (slambot_t ∗robot, double sensor_noise, double turn_noise, double motion_noise)
- void move (slambot_t ∗robot, double theta, double radius)
- void sense (slambot_t ∗robot)
- void particle_update (slambot_t ∗robot, pose3d_t ∗trajectory)
- void particle_resample (slambot_t ∗robot)
- double gauss_error (double mu, double sigma)

7.29.1 Function Documentation

7.29.1.1 double gauss_error (double *mu,* double *sigma*)

7.29.1.2 void move (slambot_t ∗ *robot,* double *theta,* double *radius*)

7.29.1.3 void particle_resample (slambot_t ∗ *robot*)

7.29.1.4 void particle_update (slambot_t ∗ *robot,* pose3d_t ∗ *trajectory*)

7.29.1.5 void sense (slambot_t ∗ *robot*)

7.29.1.6 void set_noise (slambot_t ∗ *robot,* double *sensor_noise,* double *turn_noise,* double *motion_noise*)

7.30 SEVATBR-doc/speech/speech-to-text/lse_stt.c File Reference

```
#include <stdio.h>
#include <stdint.h>
#include <stdlib.h>
#include <fcntl.h>
#include <string.h>
#include "lse_stt.h"
```

Classes

- union note

Macros

- #define NTARGETS 2

Functions

- static int16_t ∗ loadFile (char ∗fname, long ∗s)
- static unsigned long long square_error (int16_t ∗haystack, int hlen, int16_t ∗needle, int nlen, int ∗start)
- int stt_init (stt_t ∗info)
- int stt_decipher (stt_t ∗info, char ∗fname, char ∗∗buf)
- void stt_free (stt_t ∗info)

Variables

- static char ∗ db [NTARGETS]
- static long length [NTARGETS]
- static int16_t ∗ dbfiles [NTARGETS]
- static char ∗ text [NTARGETS]
- static char goodtogo

7.30.1 Macro Definition Documentation

7.30.1.1 #define NTARGETS 2

7.30.2 Function Documentation

7.30.2.1 int16_t ∗ loadFile (char ∗ *fname*, long ∗ *s*) `[static]`

Load a file from a filename, and decode into 16000HZ, 16 bit signed little endian integers

Parameters

fname	name of the raw file
s	the changable size variable

Returns

the array (malloc'd) of notes, else NULL

7.30.2.2 unsigned long long square_error (int16_t ∗ *haystack*, int *hlen*, int16_t ∗ *needle*, int *nlen*, int ∗ *start*) `[static]`

Get the least square error (lse) of the file to the dbfile comparison

Parameters

haystack	the dbfile
hlen	length of the haystack
needle	the file inputted for lse comparison
nlen	the length of the needle
start	the changable pointer to the beginning of the array

Returns

the lse on success, else max error (uint64_t)-1

7.30.2.3 int stt_decipher (stt_t ∗ *info*, char ∗ *fname*, char ∗∗ *buf*)

Decipher the current file for some hypothesis

Parameters

info	info for the lse_stt engine
fname	the name of the current file
buf	the buffer to store the deciphered hypothesis

Returns

length of the deciphered hypothesis on success, else -1

7.30.2.4 void stt_free (stt_t ∗ *info*)

Free the current engine

Parameters

info	info for the lse_stt engine

7.30.2.5 int stt_init (stt_t ∗ *info*)

Initialize the speech engine for the lse matching.

Parameters

info	info for the lse_stt engine

Returns

0 on success, -1 on error

7.30.3 Variable Documentation

7.30.3.1 char∗ db[NTARGETS] `[static]`

7.30.3.2 int16_t∗ dbfiles[NTARGETS] `[static]`

7.30.3.3 char goodtogo `[static]`

7.30.3.4 long length[NTARGETS] `[static]`

7.30.3.5 char∗ text[NTARGETS] `[static]`

7.31 SEVATBR-doc/speech/speech-to-text/lse_stt.h File Reference

Classes

- struct stt

Typedefs

- typedef struct stt stt_t

Functions

- int stt_init (stt_t ∗info)
- int stt_decipher (stt_t ∗info, char ∗filename, char ∗∗buf)
- void stt_free (stt_t ∗info)

7.31.1 Typedef Documentation

7.31.1.1 typedef struct stt stt_t

7.31.2 Function Documentation

7.31.2.1 int stt_decipher (stt_t ∗ *info,* char ∗ *filename,* char ∗∗ *buf*)

Decipher the current file for some hypothesis

Parameters

info	info for the lse_stt engine
fname	the name of the current file
buf	the buffer to store the deciphered hypothesis

Returns

length of the deciphered hypothesis on success, else -1

Try to decipher a file

Parameters

info	the information struct for the engine

Returns

n characters deciphered on success, -1 otherwise

7.31.2.2 void stt_free (stt_t ∗ *info*)

Free the current engine

Parameters

info	info for the lse_stt engine

Remove the speech engine

Parameters

info	the information struct for the engine

7.31.2.3 int stt_init (stt_t ∗ *info*)

Initialize the speech engine for the lse matching.

Parameters

info	info for the lse_stt engine

Returns

0 on success, -1 on error

Initialize the speech engine

Parameters

info	the information struct for the engine

Returns

0 on success, -1 otherwise

7.32 SEVATBR-doc/speech/speech-to-text/rawrec.c File Reference

```
#include <gst/gst.h>
#include <stdlib.h>
#include <string.h>
#include "rawrec.h"
```

Functions

- int start_recording (rawrec_t ∗rr, char ∗fsinkloc)
- void stop_recording (rawrec_t ∗rr)

Variables

- static unsigned char gst_initiallized

7.32.1 Function Documentation

7.32.1.1 int start_recording (**rawrec_t** ∗ *rr,* char ∗ *fsinkloc*)

Start the recording of the audio device.

Parameters

rr	the struct containing the necessary information for recording
fsinkloc	the filename of the destination

Returns

0 on success, -1 on error

7.32.1.2 void stop_recording (**rawrec_t** ∗ *rr*)

Stop recording of the audio device.

Parameters

rr	the struct of the pipeline information

7.32.2 Variable Documentation

7.32.2.1 unsigned char gst_initiallized [static]

7.33 SEVATBR-doc/speech/speech-to-text/rawrec.h File Reference

Classes

- struct rawrec

Typedefs

- typedef struct rawrec rawrec_t

Functions

- int start_recording (rawrec_t ∗rr, char ∗fsinkloc)
- void stop_recording (rawrec_t ∗rr)

7.33.1 Typedef Documentation

7.33.1.1 typedef struct rawrec rawrec_t

7.33.2 Function Documentation

7.33.2.1 int start_recording (rawrec_t ∗ *rr*, char ∗ *fsinkloc*)

Start the recording of the audio device.

Parameters

rr	the struct containing the necessary information for recording
fsinkloc	the filename of the destination

Returns

0 on success, -1 on error

7.33.2.2 void stop_recording (rawrec_t ∗ *rr*)

Stop recording of the audio device.

Parameters

rr	the struct of the pipeline information

7.34 SEVATBR-doc/speech/speech-to-text/speech_signal.c File Reference

```
#include <pthread.h>
#include <unistd.h>
#include <sys/wait.h>
#include <sys/types.h>
#include <stdlib.h>
#include <signal.h>
#include <string.h>
#include <stdio.h>
#include <fcntl.h>
#include "rawrec.h"
#include "stt.h"
#include "speech_signal.h"
```

Macros

- #define READPROG "sample_raw.sh"

Functions

- static void ∗ _update_signals (void ∗)
- int start_speech_signals (void)
- void get_signal (speech_signal_t ∗sigframe)
- void stop_speech_signals (void)

Variables

- static unsigned char sslib_init
- static unsigned char sslib_exit
- static stt_t sre
- static pthread_t gst_manager
- static pthread_mutex_t gst_lock
- static speech_signal_t gst_sig

7.34.1 Macro Definition Documentation

7.34.1.1 #define READPROG "sample_raw.sh"

7.34.2 Function Documentation

7.34.2.1 static void ∗ _update_signals (void ∗ *args*) [static]

Start two streams to record audio.

Parameters

args	Does nothing.

Returns

NULL

7.34.2.2 void get_signal (speech_signal_t ∗ *sigframe*)

Gets the current state of signals for the sigframe

Parameters

sigframe	the sigframe to send over in order to copy the signals

7.34.2.3 int start_speech_signals (void)

Start recording

7.34.2.4 void stop_speech_signals (void)

Stop recording

7.34.3 Variable Documentation

7.34.3.1 pthread_mutex_t gst_lock [static]

7.34.3.2 pthread_t gst_manager [static]

7.34.3.3 speech_signal_t gst_sig [static]

7.34.3.4 stt_t sre [static]

7.34.3.5 unsigned char sslib_exit [static]

7.34.3.6 unsigned char sslib_init [static]

7.35 SEVATBR-doc/speech/speech-to-text/speech_signal.h File Reference

Classes

- struct speech_signal

Typedefs

- typedef struct speech_signal speech_signal_t

Functions

- int start_speech_signals (void)
- void get_signal (speech_signal_t ∗sigframe)
- void stop_speech_signals (void)

7.35.1 Typedef Documentation

7.35.1.1 typedef struct speech_signal speech_signal_t

7.35.2 Function Documentation

7.35.2.1 void get_signal (speech_signal_t ∗ *sigframe*)

Gets the current state of signals for the sigframe

Parameters

sigframe	the sigframe to send over in order to copy the signals

7.35.2.2 int start_speech_signals (void)

Start recording

7.35.2.3 void stop_speech_signals (void)

Stop recording

7.36 SEVATBR-doc/speech/speech-to-text/stt.c File Reference

```
#include "stt.h"
#include <stdio.h>
#include <string.h>
#include <signal.h>
```

Functions

- int stt_init (stt_t ∗info)
- int stt_decipher (stt_t ∗info, char ∗filename, char ∗∗buf)
- void stt_free (stt_t ∗info)

7.36.1 Function Documentation

7.36.1.1 int stt_decipher (stt_t ∗ *info,* char ∗ *filename,* char ∗∗ *buf*)

Try to decipher a file

Parameters

info	the information struct for the engine

Returns

n characters deciphered on success, -1 otherwise

7.36.1.2 void stt_free (stt_t ∗ *info*)

Remove the speech engine

Parameters

info	the information struct for the engine

7.36.1.3 int stt_init (stt_t ∗ *info*)

Initialize the speech engine

Parameters

info	the information struct for the engine

Returns

0 on success, -1 otherwise

7.37 SEVATBR-doc/speech/speech-to-text/stt.h File Reference

```
#include <pocketsphinx.h>
```

Classes

- struct stt

Typedefs

- typedef struct stt stt_t

Functions

- int stt_init (stt_t ∗info)
- int stt_decipher (stt_t ∗info, char ∗filename, char ∗∗buf)
- void stt_free (stt_t ∗info)

7.37.1 Typedef Documentation

7.37.1.1 typedef struct stt stt_t

7.37.2 Function Documentation

7.37.2.1 int stt_decipher (stt_t ∗ *info,* char ∗ *filename,* char ∗∗ *buf*)

Decipher the current file for some hypothesis

Parameters

info	info for the lse_stt engine
fname	the name of the current file
buf	the buffer to store the deciphered hypothesis

Returns

length of the deciphered hypothesis on success, else -1

Try to decipher a file

Parameters

info	the information struct for the engine

Returns

n characters deciphered on success, -1 otherwise

7.37.2.2 void stt_free (stt_t ∗ *info*)

Free the current engine

Parameters

info	info for the lse_stt engine

Remove the speech engine

Parameters

info	the information struct for the engine

7.37.2.3 int stt_init (stt_t * *info*)

Initialize the speech engine for the lse matching.

Parameters

info	info for the lse_stt engine

Returns

0 on success, -1 on error

Initialize the speech engine

Parameters

info	the information struct for the engine

Returns

0 on success, -1 otherwise

7.38 SEVATBR-doc/visual/basket.py File Reference

Namespaces

- basket

Functions

- def basket._basket_image_hue_filter

  Internal wrapper image hue filter.
- def basket._save_image

  Saves an image to the current directory.
- def basket._init_particle_filter

  Internal wrapper to particle filter initializer.
- def basket.is_basket_middle

  Single entry function returning True/False if basket is in the middle of the screen.
- def basket.run_middle

  Runs continuously and prints if the best detected blob is in the middle.
- def basket.run

  Runs continuously outlines best matched blob if it is in the middle.

Variables

- basket.particle_filter = None
- int basket.image_half_size = -1
- int basket.save_count = 1
- tuple basket.base_filename = datetime.now()

7.39 SEVATBR-doc/visual/basket_runner.py File Reference

Namespaces

- basket_runner

7.40 SEVATBR-doc/visual/basket_test.py File Reference

Namespaces

- basket_test

Functions

- def basket_test.unitTest
- def basket_test.basketPresent
- def basket_test.basketMissing

7.41 SEVATBR-doc/visual/experiment.py File Reference

Namespaces

- experiment

Functions

- def experiment.experiment
- def experiment.hard_threshold
- def experiment.binary_mask
- def experiment.dilation_and_blur
- def experiment.blobs_by_mask

7.42 SEVATBR-doc/visual/image_support.py File Reference

Namespaces

- image_support

Functions

- def image_support.external_init_particle_filter

  Initializes particle filter.
- def image_support.image_hue_filter

  Converts given image to HSV based on the given color.
- def image_support.get_hue_blobs

  Gets basket blobs after hue distance filtering.
- def image_support.get_best_blob

  Returns the best blob out of the provided set and particle filter.
- def image_support.is_blob_in_middle_helper

  Determines whether the given blob is in ceter of image.

7.43 SEVATBR-doc/visual/particlefilter.py File Reference

Classes

- class particlefilter.ParticleFilter

Namespaces

- particlefilter

7.44 SEVATBR-doc/visual/prquadtree.py File Reference

Classes

- class prquadtree.Point

  Represents an (x,y) coordinate point on a grid.
- class prquadtree.Particle

  Represents particle point.
- class prquadtree.Box

  Class defining a square on the coordinate system via a center point and half of square width.
- class prquadtree.PRQuadTree

  Class representing a Point Range Quadtree.

Namespaces

- prquadtree

7.45 SEVATBR-doc/visual/prquadtree_test.py File Reference

Classes

- class prquadtree_test.TestPoint
- class prquadtree_test.TestParticle
- class prquadtree_test.TestBox
- class prquadtree_test.TestPrQuadTree

Namespaces

- prquadtree_test

7.46 SEVATBR-doc/visual/prquadtree_test_example.py File Reference

Namespaces

- prquadtree_test_example

Variables

- tuple prquadtree_test_example.b = Box(Point(5,5), 50)
- tuple prquadtree_test_example.b2 = Box(Point(50,50), 50)
- tuple prquadtree_test_example.qt = PRQuadTree(b2)
- tuple prquadtree_test_example.pt = Point(2,2)
- tuple prquadtree_test_example.nearby = qt.query_k_nearest(pt, 20)
- int prquadtree_test_example.c = 1

7.47 SEVATBR-doc/visual/tennis_ball.py File Reference

Namespaces

- tennis_ball

Functions

- def tennis_ball._init_particle_filter

  Internal wrapper to particle filter initializer.
- def tennis_ball._ball_image_hue_filter

  Internal wrapper image hue filter.
- def tennis_ball.is_ball_middle

  Entry point for module which determines whether tennis ball is in the middle of the image.
- def tennis_ball.run

  Continuously captures image from computer camera and feeds it to the is_ball_middle method to detect whether tennis ball is in the middle of the screen.

Variables

- tennis_ball.particle_filter = None

7.48 SEVATBR-doc/visual/tennis_ball_runner.py File Reference

Namespaces

- tennis_ball_runner

7.49 SEVATBR-doc/visual/tennis_ball_test.py File Reference

Namespaces

- tennis_ball_test

Functions

- def tennis_ball_test.unitTest
- def tennis_ball_test.ballPresent
- def tennis_ball_test.ballMissing

7.50 SEVATBR-doc/visual/visual.h File Reference

Functions

- int start_visual (void)
- void set_objects (object_t ∗objs)
- void get_objects (object_t ∗objs, point_t ∗locations)
- void stop_visual (void)

7.50.1 Function Documentation

7.50.1.1 void get_objects (object_t ∗ *objs,* point_t ∗ *locations*)

7.50.1.2 void set_objects (object_t ∗ *objs*)

7.50.1.3 int start_visual (void)

7.50.1.4 void stop_visual (void)

Index