Major Project Documentation

0.2

Generated by Doxygen 1.8.6

Mon Mar 31 2014 21:39:11

Contents

Chapter 1

README

Mayor Project Documentation

Chapter 2

Data Structure Index

2.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

Chapter 4

Data Structure Documentation

4.1 odometryTrackStruct Struct Reference

`#include <odometry.h>`

Data Fields

- struct {
  float **wheel_distance**
  float **wheel_conversion**
  } **configuration**

- struct {
  int **pos_left_prev**
  int **pos_right_prev**
  } **state**

- struct {
  float **x**
  float **y**
  float **theta**
  } **result**

4.1.1 Field Documentation

4.1.1.1 struct { ... } configuration

4.1.1.2 int pos_left_prev

4.1.1.3 int pos_right_prev

4.1.1.4 struct { ... } result

4.1.1.5 struct { ... } state

4.1.1.6 float theta

4.1.1.7 float wheel_conversion

4.1.1.8 float wheel_distance

4.1.1.9 float x

4.1.1.10 float y

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

- C:/Users/Stefan/Documents/GitHub/Final_Project/lib/**odometry.h**

Chapter 5

File Documentation

5.1 C:/Users/Stefan/Documents/GitHub/Final_Project/controllers/e_puck_movement/build/release/e-_puck_movement.d File Reference

5.2 C:/Users/Stefan/Documents/GitHub/Final_Project/controllers/Main/build/release/e_-puck_movement.d File Reference

5.3 C:/Users/Stefan/Documents/GitHub/Final_Project/controllers/mid_term_demo/build/release/e-_puck_movement.d File Reference

5.4 C:/Users/Stefan/Documents/GitHub/Final_Project/lib/build/release/e_puck_movement.d File Reference

5.5 C:/Users/Stefan/Documents/GitHub/Final_Project/controllers/epuck_collision_avoidance/build/relea_collision_avoidance.d File Reference

5.6 C:/Users/Stefan/Documents/GitHub/Final_Project/controllers/epuck_collision_avoidance/epuck-_collision_avoidance.c File Reference

```
#include <webots/robot.h>
#include <webots/differential_wheels.h>
#include <webots/distance_sensor.h>
```

Macros

- #define **TIME_STEP** 64

Functions

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

5.6.1 Macro Definition Documentation

5.6.1.1 #define TIME_STEP 64

5.6.2 Function Documentation

5.6.2.1 int main (int *argc,* char ** *argv*)

5.7 C:/Users/Stefan/Documents/GitHub/Final_Project/controllers/epuck_scanline/build/release/epuck_scanline.d File Reference

5.8 C:/Users/Stefan/Documents/GitHub/Final_Project/controllers/epuck_scanline/epuck_scanline.c File Reference

```
#include <webots/robot.h>
#include <webots/distance_sensor.h>
#include <webots/differential_wheels.h>
#include <stdlib.h>
#include <time.h>
#include <stdio.h>
```

Macros

- #define **THRESHOLD_DIST** 300
- #define **TIME_STEP** 32
- #define **SIMULATION** 0
- #define **REALITY** 2
- #define **LEFT** 0
- #define **RIGHT** 1
- #define **NB_DIST_SENS** 8
- #define **PS_RIGHT_10** 0
- #define **PS_RIGHT_45** 1
- #define **PS_RIGHT_90** 2
- #define **PS_RIGHT_REAR** 3
- #define **PS_LEFT_REAR** 4
- #define **PS_LEFT_90** 5
- #define **PS_LEFT_45** 6
- #define **PS_LEFT_10** 7
- #define **WHEEL_RADIUS** 0.02
- #define **AXLE_LENGTH** 0.026
- #define **ENCODER_RESOLUTION** 159.23
- #define **FORWARD** 0
- #define **STOP** 1
- #define **UTURN** 2
- #define **TURNRIGHT** 3
- #define **TURNLEFT** 4

Functions

- int **main** ()

Variables

- WbDeviceTag **ps** [**NB_DIST_SENS**]
- int **ps_value** [**NB_DIST_SENS**] ={0,0,0,0,0,0,0,0}
- int **ps_offset_sim** [**NB_DIST_SENS**] = {35,35,35,35,35,35,35,35}
- int **ps_offset_real** [**NB_DIST_SENS**] = {375,158,423,682,447,594,142,360}
- int **obstacle** [**NB_DIST_SENS**]
- int **speed** [2] = {0,0}
- int **state** = **FORWARD**
- int **n** =0
- int **old_encoder** =0

5.8.1 Macro Definition Documentation

5.8.1.1 #define AXLE_LENGTH 0.026

5.8.1.2 #define ENCODER_RESOLUTION 159.23

5.8.1.3 #define FORWARD 0

5.8.1.4 #define LEFT 0

5.8.1.5 #define NB_DIST_SENS 8

5.8.1.6 #define PS_LEFT_10 7

5.8.1.7 #define PS_LEFT_45 6

5.8.1.8 #define PS_LEFT_90 5

5.8.1.9 #define PS_LEFT_REAR 4

5.8.1.10 #define PS_RIGHT_10 0

5.8.1.11 #define PS_RIGHT_45 1

5.8.1.12 #define PS_RIGHT_90 2

5.8.1.13 #define PS_RIGHT_REAR 3

5.8.1.14 #define REALITY 2

5.8.1.15 #define RIGHT 1

5.8.1.16 #define SIMULATION 0

5.8.1.17 #define STOP 1

5.8.1.18 #define THRESHOLD_DIST 300

5.8.1.19 #define TIME_STEP 32

5.8.1.20 #define TURNLEFT 4

5.8.1.21 #define TURNRIGHT 3

5.8.1.22 #define UTURN 2

5.8.1.23 #define WHEEL_RADIUS 0.02

5.8.2 Function Documentation

5.8.2.1 int main ( )

5.8.3 Variable Documentation

5.8.3.1 int n =0

5.8.3.2 int obstacle[NB_DIST_SENS]

5.8.3.3 int old_encoder =0

5.8.3.4 WbDeviceTag ps

5.8.3.5 int ps_offset_real[NB_DIST_SENS] = {375,158,423,682,447,594,142,360}

5.8.3.6 int ps_offset_sim[NB_DIST_SENS] = {35,35,35,35,35,35,35,35}

5.8.3.7 int ps_value[NB_DIST_SENS] ={0,0,0,0,0,0,0,0}

5.8.3.8 int speed[2] = {0,0}

5.8.3.9 int state = FORWARD

5.9 C:/Users/Stefan/Documents/GitHub/Final_Project/controllers/Main/build/release/-Main.d File Reference

5.10 C:/Users/Stefan/Documents/GitHub/Final_Project/controllers/Main/build/release/map-_building.d File Reference

5.11 C:/Users/Stefan/Documents/GitHub/Final_Project/controllers/mid_term_demo/build/release/map-_building.d File Reference

5.12 C:/Users/Stefan/Documents/GitHub/Final_Project/lib/build/release/map_building.d File Reference

5.13 C:/Users/Stefan/Documents/GitHub/Final_Project/controllers/Main/build/release/odometry.d File Reference

5.14 C:/Users/Stefan/Documents/GitHub/Final_Project/lib/build/release/odometry.d File Reference

5.15 C:/Users/Stefan/Documents/GitHub/Final_Project/controllers/Main/Main.c File Reference

```
#include <webots/robot.h>
#include <webots/differential_wheels.h>
#include <webots/distance_sensor.h>
#include <webots/light_sensor.h>
#include <webots/camera.h>
#include <webots/display.h>
#include <webots/accelerometer.h>
#include <webots/led.h>
#include <math.h>
#include "../../lib/odometry.h"
#include "../../lib/e_puck_movement.h"
#include "../../lib/map_building.h"
```

Macros

- #define **TIME_STEP** 8
- #define **MAP_SIZE** 70
- #define **THRESHOLD_DIST** 100
- #define **OCCUPANCE_DIST** 150
- #define **LEFT** 0
- #define **RIGHT** 1
- #define **NUM_DIST_SENS** 8
- #define **PS_RIGHT_10** 0
- #define **PS_RIGHT_45** 1
- #define **PS_RIGHT_90** 2
- #define **PS_RIGHT_REAR** 3
- #define **PS_LEFT_REAR** 4
- #define **PS_LEFT_90** 5
- #define **PS_LEFT_45** 6
- #define **PS_LEFT_10** 7

Functions

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

Variables

- WbDeviceTag **led** [3]
- struct **odometryTrackStruct ot**

5.15.1 Macro Definition Documentation

5.15.1.1 #define LEFT 0

5.15.1.2 #define MAP_SIZE 70

5.15.1.3 #define NUM_DIST_SENS 8

5.15.1.4 #define OCCUPANCE_DIST 150

5.15.1.5 #define PS_LEFT_10 7

5.15.1.6 #define PS_LEFT_45 6

5.15.1.7 #define PS_LEFT_90 5

5.15.1.8 #define PS_LEFT_REAR 4

5.15.1.9 #define PS_RIGHT_10 0

5.15.1.10 #define PS_RIGHT_45 1

5.15.1.11 #define PS_RIGHT_90 2

5.15.1.12 #define PS_RIGHT_REAR 3

5.15.1.13 #define RIGHT 1

5.15.1.14 #define THRESHOLD_DIST 100

5.15.1.15 #define TIME_STEP 8

5.15.2 Function Documentation

5.15.2.1 int main (int *argc,* char ** *argv*)

5.15.3 Variable Documentation

5.15.3.1 WbDeviceTag led[3]

5.15.3.2 struct odometryTrackStruct ot

5.16 C:/Users/Stefan/Documents/GitHub/Final_Project/controllers/mid_term_demo/build/release/mid_term_demo.d File Reference

5.17 C:/Users/Stefan/Documents/GitHub/Final_Project/controllers/mid_term_demo/mid_term_demo.c File Reference

```
#include <webots/robot.h>
#include "../../lib/map_building.h"
#include "../../lib/e_puck_movement.h"
```

Macros

- #define **TIME_STEP** 8

Functions

- int **main** (int argc, char **argv)

5.17.1 Macro Definition Documentation

5.17.1.1 #define TIME_STEP 8

5.17.2 Function Documentation

5.17.2.1 int main (int *argc,* char ** *argv*)

5.18 C:/Users/Stefan/Documents/GitHub/Final_Project/controllers/my_controller_epuk_-go_forward/build/release/my_controller_epuk_go_forward.d File Reference

5.19 C:/Users/Stefan/Documents/GitHub/Final_Project/controllers/my_controller_epuk_-go_forward/my_controller_epuk_go_forward.c File Reference

```
#include <webots/robot.h>
#include <webots/differential_wheels.h>
```

Macros

- #define **TIME_STEP** 64

Functions

- int **main** (int argc, char **argv)

5.19.1 Macro Definition Documentation

5.19.1.1 #define TIME_STEP 64

5.19.2 Function Documentation

5.19.2.1 int main (int *argc,* char ** *argv*)

5.20 C:/Users/Stefan/Documents/GitHub/Final_Project/epuck‘_movement_download.c File Reference

```
#include <webots/robot.h>
#include <webots/differential_wheels.h>
#include <webots/distance_sensor.h>
#include <webots/light_sensor.h>
#include <webots/camera.h>
#include <webots/accelerometer.h>
#include <webots/led.h>
#include <math.h>
```

Macros

- #define **M_PI** 3.14159265358979323846264338327 95L

- #define **TIME_STEP** 8
- #define **WHEEL_RADIUS** 0.0206625
- #define **WHEELBASE** 0.052
- #define **ENCODER_RESOLUTION** 159.23
- #define **REV_STEP** 1000
- #define **STEP_TOLERANCE** 6.0
- #define **RANGE** (1024 / 2)
- #define **MIN_DIST** 20.0f
- #define **MIN_SPEED** 10.0f
- #define **NTOURNAMENTS** 1

Functions

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

Variables

- double **dMSpeed** [2] = {0.0f, 0.0f}
- double **dPrevEncPos** [2] = {0.0f, 0.0f}
- double ∗ **p_dOdometryData**
- WbDeviceTag **led** [3]

5.20.1 Macro Definition Documentation

5.20.1.1 #define ENCODER_RESOLUTION 159.23

5.20.1.2 #define M_PI 3.14159265358979323846264338327 95L

5.20.1.3 #define MIN_DIST 20.0f

5.20.1.4 #define MIN_SPEED 10.0f

5.20.1.5 #define NTOURNAMENTS 1

5.20.1.6 #define RANGE (1024 / 2)

5.20.1.7 #define REV_STEP 1000

5.20.1.8 #define STEP_TOLERANCE 6.0

5.20.1.9 #define TIME_STEP 8

5.20.1.10 #define WHEEL_RADIUS 0.0206625

5.20.1.11 #define WHEELBASE 0.052

5.20.2 Function Documentation

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

5.20.3 Variable Documentation

5.20.3.1 double dMSpeed[2] = {0.0f, 0.0f}

5.20.3.2 double dPrevEncPos[2] = {0.0f, 0.0f}

5.20.3.3 WbDeviceTag led[3]

5.20.3.4 double* p_dOdometryData

5.21 C:/Users/Stefan/Documents/GitHub/Final_Project/lib/e_puck_movement.c File Reference

```
#include <webots/robot.h>
#include <webots/differential_wheels.h>
#include <webots/distance_sensor.h>
#include <webots/light_sensor.h>
#include <webots/camera.h>
#include <webots/accelerometer.h>
#include <webots/led.h>
#include <math.h>
#include "e_puck_movement.h"
#include "map_building.h"
#include "odometry.h"
```

Macros

- #define **M_PI** 3.14159265358979323846264338327 95L
- #define **TIME_STEP** 8
- #define **WHEEL_RADIUS** 0.0206625
- #define **WHEELBASE** 0.052
- #define **ENCODER_RESOLUTION** 159.23
- #define **INCREMENT_STEP** 1000
- #define **STEP_TOLERANCE** 6.0
- #define **RANGE** (1024 / 2)
- #define **MIN_DIST** 20.0f
- #define **MIN_SPEED** 10.0f
- #define **NUMTOURNAMENTS** 1
- #define **RTOD**(r) ((r) * 180 / **M_PI**)
- #define **ANGLE_TOLERANCE** 5
- #define **EAST** 0
- #define **NORTH** 90
- #define **WEST** 180
- #define **SOUTH** 270

Functions

- void **stop_robot** ()
- void **move_forward** (double dSpeed, double dDist)
- void **turn_left** (double dSpeed)
- void **turn_right** (double dSpeed)
- void **turn_angle** (double dAngle, double dSpeed)
- void **set_motor_speed** (double dSpeedL, double dSpeedR)
- double * **get_encoder_positions** ()
- void **check_rotation** (double cur_rot, double want_rot, double dSpeed)
- double * **compute_odometry_data** ()

- void **UMBmark** (double dSpeed, double dDistance)
- void **measure_clockWise** (double dSpeed, double dDistance)
- void **measure_CounterClockWise** (double dSpeed, double dDistance)
- void **set_leds** (int iActive)

Variables

- double **dCurSpeed** [2] = {0.0f, 0.0f}
- double **dPrevEncPos** [2] = {0.0f, 0.0f}
- double * **point_dOdometryData**
- WbDeviceTag **led** [3]
- float **weights_left** [8] = {-1,-1,-1,0.5,-0.5,0.5,1,2}
- float **weights_right** [8] = {1,0.8,1,-0.5,0.5,-1,-1.6,-2}

5.21.1 Macro Definition Documentation

5.21.1.1 #define ANGLE_TOLERANCE 5

5.21.1.2 #define EAST 0

5.21.1.3 #define ENCODER_RESOLUTION 159.23

5.21.1.4 #define INCREMENT_STEP 1000

5.21.1.5 #define M_PI 3.14159265358979323846264338327 95L

5.21.1.6 #define MIN_DIST 20.0f

5.21.1.7 #define MIN_SPEED 10.0f

5.21.1.8 #define NORTH 90

5.21.1.9 #define NUMTOURNAMENTS 1

5.21.1.10 #define RANGE (1024 / 2)

5.21.1.11 #define RTOD(*r*) ((r) * 180 / M_PI)

5.21.1.12 #define SOUTH 270

5.21.1.13 #define STEP_TOLERANCE 6.0

5.21.1.14 #define TIME_STEP 8

5.21.1.15 #define WEST 180

5.21.1.16 #define WHEEL_RADIUS 0.0206625

5.21.1.17 #define WHEELBASE 0.052

5.21.2 Function Documentation

5.21.2.1 void check_rotation (double *cur_rot,* double *want_rot,* double *dSpeed*)

FUnction to compare the current heading to the wanted heading and fix the heading should it surpass a threshold

5.21.2.2 double * compute_odometry_data ( )

Function to compute the robots current odometry data this includes the x and y placement as well as the rotation theta

5.21.2.3 double * get_encoder_positions ( )

Function to get and return the current encoder position of the wheels

5.21.2.4 void measure_clockWise (double *dSpeed,* double *dDistance*)

Function to measure the movement accuracy by driving a clockwise square. This is part of the UMBmark algorithm

5.21.2.5 void measure_CounterClockWise (double *dSpeed,* double *dDistance*)

Function to measure the movement accuracy by driving a counter-clockwise square. This is part of the UMBmark algorithm

5.21.2.6 void move_forward (double *dSped,* double *dDis*)

Function to move the robot forward a given distance at a given speed

5.21.2.7 void set_leds (int *iActive*)

set the status of the LEDs

5.21.2.8 void set_motor_speed (double *dLeftSpeed,* double *dRightSpeed*)

Function to set the motor speed of the robot

5.21.2.9 void stop_robot ( )

Function to stop the robot

5.21.2.10 void turn_angle (double *dAngle,* double *dSpeed*)

Function to turn the robot a given angle with a given speed

5.21.2.11 void turn_left (double *dSpeed*)

Function to turn left

5.21.2.12 void turn_right (double *dSpeed*)

Function to turn right

5.21.2.13 void UMBmark (double *dSpeed,* double *dDistance*)

University of Michigan Benchmark

5.21.3 Variable Documentation

5.21.3.1 double dCurSpeed[2] = {0.0f, 0.0f}

5.21.3.2 double dPrevEncPos[2] = {0.0f, 0.0f}

5.21.3.3 WbDeviceTag led[3]

5.21.3.4 double* point_dOdometryData

5.21.3.5 float weights_left[8] = {-1,-1,-1,0.5,-0.5,0.5,1,2}

5.21.3.6 float weights_right[8] = {1,0.8,1,-0.5,0.5,-1,-1.6,-2}

5.22 C:/Users/Stefan/Documents/GitHub/Final_Project/lib/e_puck_movement.h File Reference

Functions

- void **stop_robot** ()
- void **move_forward** (double dSpeed, double dDist)
- void **turn_left** (double dSpeed)
- void **turn_right** (double dSpeed)
- void **turn_angle** (double dAngle, double dSpeed)
- void **set_motor_speed** (double dSpeedL, double dSpeedR)
- double * **get_encoder_positions** ()
- double * **compute_odometry_data** ()
- void **UMBmark** (double dSpeed, double dDistance)
- void **measure_clockWise** (double dSpeed, double dDistance)
- void **measure_CounterClockWise** (double dSpeed, double dDistance)
- void **set_leds** (int iActive)
- void **controll_angle** ()
- void **check_rotation** (double cur_rot, double want_rot, double dSpeed)

5.22.1 Function Documentation

5.22.1.1 void check_rotation (double *cur_rot,* double *want_rot,* double *dSpeed*)

FUnction to compare the current heading to the wanted heading and fix the heading should it surpass a threshold

5.22.1.2 double* compute_odometry_data ( )

Function to compute the robots current odometry data this includes the x and y placement as well as the rotation theta

5.22.1.3 void controll_angle ( )

5.22.1.4 double* get_encoder_positions ( )

Function to get and return the current encoder position of the wheels

5.22.1.5 void measure_clockWise (double *dSpeed,* double *dDistance*)

Function to measure the movement accuracy by driving a clockwise square. This is part of the UMBmark algorithm

5.22.1.6 void measure_CounterClockWise (double *dSpeed,* double *dDistance*)

Function to measure the movement accuracy by driving a counter-clockwise square. This is part of the UMBmark algorithm

5.22.1.7 void move_forward (double *dSpeed,* double *dDist*)

Function to move the robot forward a given distance at a given speed

5.22.1.8 void set_leds (int *iActive*)

set the status of the LEDs

5.22.1.9 void set_motor_speed (double *dSpeedL,* double *dSpeedR*)

Function to set the motor speed of the robot

5.22.1.10 void stop_robot ( )

Function to stop the robot

5.22.1.11 void turn_angle (double *dAngle,* double *dSpeed*)

Function to turn the robot a given angle with a given speed

5.22.1.12 void turn_left (double *dSpeed*)

Function to turn left

5.22.1.13 void turn_right (double *dSpeed*)

Function to turn right

5.22.1.14 void UMBmark (double *dSpeed,* double *dDistance*)

University of Michigan Benchmark

5.23 C:/Users/Stefan/Documents/GitHub/Final_Project/lib/map_building.c File Reference

```
#include <webots/robot.h>
#include <webots/differential_wheels.h>
#include <webots/distance_sensor.h>
#include <webots/light_sensor.h>
#include <math.h>
#include <webots/display.h>
#include <stdio.h>
#include "map_building.h"
#include "e_puck_movement.h"
#include "odometry.h"
```

Macros

- #define **TIME_STEP** 8
- #define **MAP_SIZE** 70
- #define **CELL_SIZE** 0.015
- #define **THRESHOLD_DIST** 100
- #define **OCCUPANCE_DIST** 150
- #define **LEFT** 0
- #define **RIGHT** 1
- #define **NUM_DIST_SENS** 8
- #define **PS_RIGHT_10** 0
- #define **PS_RIGHT_45** 1
- #define **PS_RIGHT_90** 2
- #define **PS_RIGHT_REAR** 3
- #define **PS_LEFT_REAR** 4
- #define **PS_LEFT_90** 5
- #define **PS_LEFT_45** 6
- #define **PS_LEFT_10** 7
- #define **FORWARD** 0
- #define **STOP** 1
- #define **TURNRIGHT** 2
- #define **TURNLEFT** 3
- #define **UTURN** 4
- #define **RTOD**(r) ((r) ∗ 180 / **M_PI**)
- #define **ANGLE_TOLERANCE** 20
- #define **EAST** 0
- #define **NORTH** 90
- #define **WEST** 180
- #define **SOUTH** 270

Functions

- void **check_direction** (double d)
- void **init_display** ()
- void **occupied_cell** (int x, int y, float theta)
- void **reset** ()
- void **run** (struct **odometryTrackStruct** ∗**ot**)
- int ∗ **return_sensor_values** ()
- int **return_angle** (double rad)

Variables

- WbDeviceTag **led** [3]
- WbDeviceTag **ps** [**NUM_DIST_SENS**]
- WbDeviceTag **display**
- WbImageRef **background**
- int **display_width**
- int **display_height**
- int **map** [**MAP_SIZE**][**MAP_SIZE**]
- int **robot_x** = **MAP_SIZE** / 2
- int **robot_y** = **MAP_SIZE** / 2
- int **ps_value** [**NUM_DIST_SENS**] ={0,0,0,0,0,0,0,0}
- int **obstacle** [**NUM_DIST_SENS**] ={0,0,0,0,0,0,0,0}
- bool **ob_front**
- bool **ob_right**
- bool **ob_left**
- float **angle_offset** [**NUM_DIST_SENS**] = {0.2793, 0.7854, 1.5708, 2.618, -2.618, -1.5708, -0.7854, -0.2793}
- int **new_encoder**
- bool **north**
- bool **west**
- bool **south**
- bool **east**
- int **state** = **FORWARD**

5.23.1 Macro Definition Documentation

5.23.1.1 #define ANGLE_TOLERANCE 20

5.23.1.2 #define CELL_SIZE 0.015

5.23.1.3 #define EAST 0

5.23.1.4 #define FORWARD 0

5.23.1.5 #define LEFT 0

5.23.1.6 #define MAP_SIZE 70

5.23.1.7 #define NORTH 90

5.23.1.8 #define NUM_DIST_SENS 8

5.23.1.9 #define OCCUPANCE_DIST 150

5.23.1.10 #define PS_LEFT_10 7

5.23.1.11 #define PS_LEFT_45 6

5.23.1.12 #define PS_LEFT_90 5

5.23.1.13 #define PS_LEFT_REAR 4

5.23.1.14 #define PS_RIGHT_10 0

5.23.1.15 #define PS_RIGHT_45 1

5.23.1.16 #define PS_RIGHT_90 2

5.23.1.17 #define PS_RIGHT_REAR 3

5.23.1.18 #define RIGHT 1

5.23.1.19 #define RTOD(*r*) ((r) * 180 / M_PI)

5.23.1.20 #define SOUTH 270

5.23.1.21 #define STOP 1

5.23.1.22 #define THRESHOLD_DIST 100

5.23.1.23 #define TIME_STEP 8

5.23.1.24 #define TURNLEFT 3

5.23.1.25 #define TURNRIGHT 2

5.23.1.26 #define UTURN 4

5.23.1.27 #define WEST 180

5.23.2 Function Documentation

5.23.2.1 void check_direction (double *d*)

set booleans for the direction the robot is moving in

5.23.2.2 void init_display ( )

Initiate the display with a white color

5.23.2.3 void occupied_cell (int *x*, int *y*, float *theta*)

Set the coresponding cell to 1 (occuM_PIed) and display it

5.23.2.4 void reset (void)

enables the needed sensor devices

5.23.2.5 int return_angle (double *rad*)

returns the angle in which the robot is moving

5.23.2.6 int* return_sensor_values ( )

returns a pointer to an array of the current sensor values

5.23.2.7 void run (struct odometryTrackStruct * *ot*)

run function

5.23.3 Variable Documentation

5.23.3.1 float angle_offset[NUM_DIST_SENS] = {0.2793, 0.7854, 1.5708, 2.618, -2.618, -1.5708, -0.7854, -0.2793}

5.23.3.2 WbImageRef background

5.23.3.3 WbDeviceTag display

5.23.3.4 int display_height

5.23.3.5 int display_width

5.23.3.6 bool east

5.23.3.7 WbDeviceTag led[3]

5.23.3.8 int map[MAP_SIZE][MAP_SIZE]

5.23.3.9 int new_encoder

5.23.3.10 bool north

5.23.3.11 bool ob_front

5.23.3.12 bool ob_left

5.23.3.13 bool ob_right

5.23.3.14 int obstacle[NUM_DIST_SENS] ={0,0,0,0,0,0,0,0}

5.23.3.15 WbDeviceTag ps[NUM_DIST_SENS]

5.23.3.16 int ps_value[NUM_DIST_SENS] ={0,0,0,0,0,0,0,0}

5.23.3.17 int robot_x = MAP_SIZE / 2

5.23.3.18 int robot_y = MAP_SIZE / 2

5.23.3.19 bool south

5.23.3.20 int state = FORWARD

5.23.3.21 bool west

5.24 C:/Users/Stefan/Documents/GitHub/Final_Project/map_building.c File Reference

```
#include <webots/robot.h>
#include <webots/differential_wheels.h>
#include <webots/distance_sensor.h>
#include <webots/light_sensor.h>
#include <math.h>
#include <webots/display.h>
#include "map_building.h"
#include "e_puck_movement.h"
#include "odometry.h"
```

Macros

- #define **TIME_STEP** 8
- #define **MAP_SIZE** 70
- #define **CELL_SIZE** 0.015
- #define **THRESHOLD_DIST** 100
- #define **OCCUPANCE_DIST** 150
- #define **LEFT** 0
- #define **RIGHT** 1
- #define **NUM_DIST_SENS** 8
- #define **PS_RIGHT_10** 0
- #define **PS_RIGHT_45** 1
- #define **PS_RIGHT_90** 2
- #define **PS_RIGHT_REAR** 3
- #define **PS_LEFT_REAR** 4
- #define **PS_LEFT_90** 5
- #define **PS_LEFT_45** 6
- #define **PS_LEFT_10** 7
- #define **FORWARD** 0
- #define **TURNRIGHT** 1
- #define **TURNLEFT** 2
- #define **UTURN** 3

Functions

- void **init_display** ()
- void **occupied_cell** (int x, int y, float theta)
- void **reset** ()
- void **run** (struct **odometryTrackStruct** ∗**ot**)
- int ∗ **return_sensor_values** ()

Variables

- WbDeviceTag **led** [3]
- WbDeviceTag **ps** [**NUM_DIST_SENS**]
- WbDeviceTag **display**
- WbImageRef **background**
- int **display_width**
- int **display_height**
- int **map** [**MAP_SIZE**][**MAP_SIZE**]
- int **robot_x** = **MAP_SIZE** / 2
- int **robot_y** = **MAP_SIZE** / 2
- int **ps_value** [**NUM_DIST_SENS**] ={0,0,0,0,0,0,0,0}
- int **obstacle** [**NUM_DIST_SENS**] ={0,0,0,0,0,0,0,0}
- bool **ob_front**
- bool **ob_right**
- bool **ob_left**
- float **angle_offset** [**NUM_DIST_SENS**] = {0.2793, 0.7854, 1.5708, 2.618, -2.618, -1.5708, -0.7854, -0.2793}
- int **new_encoder**
- int **state** = **FORWARD**

5.24.1 Macro Definition Documentation

5.24.1.1 #define CELL_SIZE 0.015

5.24.1.2 #define FORWARD 0

5.24.1.3 #define LEFT 0

5.24.1.4 #define MAP_SIZE 70

5.24.1.5 #define NUM_DIST_SENS 8

5.24.1.6 #define OCCUPANCE_DIST 150

5.24.1.7 #define PS_LEFT_10 7

5.24.1.8 #define PS_LEFT_45 6

5.24.1.9 #define PS_LEFT_90 5

5.24.1.10 #define PS_LEFT_REAR 4

5.24.1.11 #define PS_RIGHT_10 0

5.24.1.12 #define PS_RIGHT_45 1

5.24.1.13 #define PS_RIGHT_90 2

5.24.1.14 #define PS_RIGHT_REAR 3

5.24.1.15 #define RIGHT 1

5.24.1.16 #define THRESHOLD_DIST 100

5.24.1.17 #define TIME_STEP 8

5.24.1.18 #define TURNLEFT 2

5.24.1.19 #define TURNRIGHT 1

5.24.1.20 #define UTURN 3

5.24.2 Function Documentation

5.24.2.1 void init_display ( )

Initiate the display with a white color

5.24.2.2 void occupied_cell (int *x*, int *y*, float *theta*)

Set the coresponding cell to 1 (occuM_PIed) and display it

5.24.2.3 void reset (void)

enables the needed sensor devices

5.24.2.4 int* return_sensor_values ( )

returns a pointer to an array of the current sensor values

5.24.2.5 void run (struct odometryTrackStruct * *ot*)

5.24.3 Variable Documentation

5.24.3.1 float angle_offset[NUM_DIST_SENS] = {0.2793, 0.7854, 1.5708, 2.618, -2.618, -1.5708, -0.7854, -0.2793}

5.24.3.2 WbImageRef background

5.24.3.3 WbDeviceTag display

5.24.3.4 int display_height

5.24.3.5 int display_width

5.24.3.6 WbDeviceTag led[3]

5.24.3.7 int map[MAP_SIZE][MAP_SIZE]

5.24.3.8 int new_encoder

5.24.3.9 bool ob_front

5.24.3.10 bool ob_left

5.24.3.11 bool ob_right

5.24.3.12 int obstacle[NUM_DIST_SENS] ={0,0,0,0,0,0,0,0}

5.24.3.13 WbDeviceTag ps[NUM_DIST_SENS]

5.24.3.14 int ps_value[NUM_DIST_SENS] ={0,0,0,0,0,0,0,0}

5.24.3.15 int robot_x = MAP_SIZE / 2

5.24.3.16 int robot_y = MAP_SIZE / 2

5.24.3.17 int state = FORWARD

5.25 C:/Users/Stefan/Documents/GitHub/Final_Project/lib/map_building.h File Reference

Functions

- void **init_display** ()
- void **occupied_cell** (int x, int y, float theta)
- void **reset** ()
- void **run** ()
- int * **return_sensor_values** ()
- int **return_angle** ()

5.25.1 Function Documentation

5.25.1.1 void init_display ( )

Initiate the display with a white color

5.25.1.2 void occupied_cell (int *x*, int *y*, float *theta*)

Set the coresponding cell to 1 (occuM_PIed) and display it

5.25.1.3 void reset (void)

enables the needed sensor devices

5.25.1.4 int return_angle ( )

5.25.1.5 int* return_sensor_values ( )

returns a pointer to an array of the current sensor values

5.25.1.6 void run ( )

5.26 C:/Users/Stefan/Documents/GitHub/Final_Project/lib/odometry.c File Reference

```
#include <webots/differential_wheels.h>
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <math.h>
#include "odometry.h"
#include "e_puck_movement.h"
#include "errno.h"
```

Macros

- #define **M_PI** 3.14159265358979323846264338327 95L

Functions

- int **odometry_track_start** (struct **odometryTrackStruct** *__ot__)
- int **odometry_track_start_pos** (struct **odometryTrackStruct** *__ot__, double *dEncPos)
- void **odometry_track_step** (struct **odometryTrackStruct** *__ot__)
- void **odometry_track_step_pos** (struct **odometryTrackStruct** *__ot__, double *dEncPos)

Variables

- float **increments_per_tour** = 1000.0
- float **wheel_radius** = 0.0207
- float **scaling_factor** = 1
- float **axis_wheel_ratio** = 1.4134

5.26.1 Macro Definition Documentation

5.26.1.1 #define M_PI 3.14159265358979323846264338327 95L

5.26.2 Function Documentation

5.26.2.1 int odometry_track_start (struct odometryTrackStruct * ot)

Initializes the odometry algortihms

5.26.2.2 int odometry_track_start_pos (struct odometryTrackStruct * ot, double * dEncPos)

Start the odometry tracking Updates the info in the **odometryTrackStruct** (p. 7) for the first time

5.26.2.3 void odometry_track_step (struct odometryTrackStruct * ot)

5.26.2.4 void odometry_track_step_pos (struct odometryTrackStruct * ot, double * dEncPos)

5.26.3 Variable Documentation

5.26.3.1 float axis_wheel_ratio = 1.4134

5.26.3.2 float increments_per_tour = 1000.0

5.26.3.3 float scaling_factor = 1

5.26.3.4 float wheel_radius = 0.0207

5.27 C:/Users/Stefan/Documents/GitHub/Final_Project/odometry.c File Reference

```
#include <webots/differential_wheels.h>
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <math.h>
#include "odometry.h"
#include "e_puck_movement.h"
#include "errno.h"
```

Macros

- #define **M_PI** 3.14159265358979323846264338327 95L

Functions

- int **odometry_track_start** (struct **odometryTrackStruct** *ot)
- int **odometry_track_start_pos** (struct **odometryTrackStruct** *ot, double *dEncPos)
- void **odometry_track_step** (struct **odometryTrackStruct** *ot)
- void **odometry_track_step_pos** (struct **odometryTrackStruct** *ot, double *dEncPos)

Variables

- float **increments_per_tour** = 1000.0
- float **wheel_radius** = 0.0207
- float **scaling_factor** = 1
- float **axis_wheel_ratio** = 1.4134

5.27.1 Macro Definition Documentation

5.27.1.1 #define M_PI 3.14159265358979323846264338327951

5.27.2 Function Documentation

5.27.2.1 int odometry_track_start (struct **odometryTrackStruct** * *ot*)

Initializes the odometry algortihms

5.27.2.2 int odometry_track_start_pos (struct **odometryTrackStruct** * *ot,* double * *dEncPos*)

Start the odometry tracking Updates the info in the **odometryTrackStruct** (p. 7) for the first time

5.27.2.3 void odometry_track_step (struct **odometryTrackStruct** * *ot*)

5.27.2.4 void odometry_track_step_pos (struct **odometryTrackStruct** * *ot,* double * *dEncPos*)

5.27.3 Variable Documentation

5.27.3.1 float axis_wheel_ratio = 1.4134

5.27.3.2 float increments_per_tour = 1000.0

5.27.3.3 float scaling_factor = 1

5.27.3.4 float wheel_radius = 0.0207

5.28 C:/Users/Stefan/Documents/GitHub/Final_Project/lib/odometry.h File Reference

Data Structures

- struct **odometryTrackStruct**

Functions

- int **odometry_track_start** (struct **odometryTrackStruct** *__ot__)
- int **odometry_track_start_pos** (struct **odometryTrackStruct** *__ot__, double *dEncPos)
- void **odometry_track_step** (struct **odometryTrackStruct** *__ot__)
- void **odometry_track_step_pos** (struct **odometryTrackStruct** *__ot__, double *dEncPos)

5.28.1 Function Documentation

5.28.1.1 int odometry_track_start (struct **odometryTrackStruct** * *ot*)

Initializes the odometry algortihms

5.28.1.2 int odometry_track_start_pos (struct **odometryTrackStruct** * *ot*, double * *dEncPos*)

Start the odometry tracking Updates the info in the **odometryTrackStruct** (p. 7) for the first time

5.28.1.3 void odometry_track_step (struct **odometryTrackStruct** * *ot*)

5.28.1.4 void odometry_track_step_pos (struct **odometryTrackStruct** * *ot*, double * *dEncPos*)

5.29 C:/Users/Stefan/Documents/GitHub/Final_Project/README.md File Reference

5.30 C:/Users/Stefan/Documents/GitHub/Final_Project/scanline_backup.c File Reference

```
#include <webots/robot.h>
#include <webots/differential_wheels.h>
#include <webots/distance_sensor.h>
#include <stdlib.h>
#include <stdio.h>
```

Macros

- #define **TIME_STEP** 64
- #define **LEFT** 0
- #define **RIGHT** 1
- #define **THRESHOLD_DIST** 100
- #define **NUM_DIST_SENS** 8
- #define **PS_RIGHT_10** 0
- #define **PS_RIGHT_45** 1
- #define **PS_RIGHT_90** 2
- #define **PS_RIGHT_REAR** 3
- #define **PS_LEFT_REAR** 4
- #define **PS_LEFT_90** 5
- #define **PS_LEFT_45** 6
- #define **PS_LEFT_10** 7
- #define **FORWARD** 0
- #define **STOP** 1
- #define **UTURN** 2
- #define **TURNRIGHT** 3
- #define **TURNLEFT** 4
- #define **WHEEL_RADIUS** 0.02
- #define **AXLE_LENGTH** 0.026
- #define **ENCODER_RESOLUTION** 159.23

Functions

- void **fsm** ()
- int **main** ()

Variables

- WbDeviceTag **ps** [**NUM_DIST_SENS**]
- int **ps_value** [**NUM_DIST_SENS**] ={0,0,0,0,0,0,0,0}
- int **ps_offset** [**NUM_DIST_SENS**] = {35,35,35,35,35,35,35,35}
- int **obstacle** [**NUM_DIST_SENS**]
- int **state** = **FORWARD**
- int **speed** [2] = {0,0}
- int **fsm_speed** [2] = {0,0}

5.30.1 Macro Definition Documentation

5.30.1.1 #define AXLE_LENGTH 0.026

5.30.1.2 #define ENCODER_RESOLUTION 159.23

5.30.1.3 #define FORWARD 0

5.30.1.4 #define LEFT 0

5.30.1.5 #define NUM_DIST_SENS 8

5.30.1.6 #define PS_LEFT_10 7

5.30.1.7 #define PS_LEFT_45 6

5.30.1.8 #define PS_LEFT_90 5

5.30.1.9 #define PS_LEFT_REAR 4

5.30.1.10 #define PS_RIGHT_10 0

5.30.1.11 #define PS_RIGHT_45 1

5.30.1.12 #define PS_RIGHT_90 2

5.30.1.13 #define PS_RIGHT_REAR 3

5.30.1.14 #define RIGHT 1

5.30.1.15 #define STOP 1

5.30.1.16 #define THRESHOLD_DIST 100

5.30.1.17 #define TIME_STEP 64

5.30.1.18 #define TURNLEFT 4

5.30.1.19 #define TURNRIGHT 3

5.30.1.20 #define UTURN 2

5.30.1.21 #define WHEEL_RADIUS 0.02

5.30.2 Function Documentation

5.30.2.1 void fsm ( )

Function which holds the FSM

5.30.2.2 int main ( )

Main method

5.30.3 Variable Documentation

5.30.3.1 int fsm_speed[2] = {0,0}

5.30.3.2 int obstacle[NUM_DIST_SENS]

5.30.3.3 WbDeviceTag ps[NUM_DIST_SENS]

Global Values

5.30.3.4 int ps_offset[NUM_DIST_SENS] = {35,35,35,35,35,35,35,35}

5.30.3.5 int ps_value[NUM_DIST_SENS] ={0,0,0,0,0,0,0,0}

5.30.3.6 int speed[2] = {0,0}

5.30.3.7 int state = FORWARD