Porthos

An-embedded-linux-robot

0.0.0-cmake

generated on Sat Nov 19 2016 22:36:17

Contents

1	Port	thos	1
2	Lice	ensing	3
	2.1	Licensing of documentation	3
	2.2	Licensing of source code	3
3	Sys	tem Description	5
4	Test	ting	7
	4.1	Unittests	7
5	Wor	kflow	9
	5.1	Workflow	9
	5.2	Setting up	9
	5.3	Starting on your feature	9
	5.4	Finish work	10
	5.5	Cleanup	10
6	Tod	o List	11
7	File	Index	13
	7.1	File List	13
8	File	Documentation	15
	8.1	licensing.dox File Reference	15
	8.2	motion.c File Reference	15
		8.2.1 Function Documentation	15
		8.2.1.1 motion_init()	15
	8.3	motion.h File Reference	15
		8.3.1 Function Documentation	15
		8.3.1.1 motion_init()	16
	8.4	README.md File Reference	16
	8.5	system_description.dox File Reference	16
	8.6	test librotion c File Reference	16

iv CONTENTS

	8.6.1	Function	Documenta	ation	 	 	 			 					 16
		8.6.1.1	main() .		 	 	 			 					 16
		8.6.1.2	motion()		 	 	 			 					 16
		8.6.1.3	test_init()		 	 	 			 					 16
8.7	testing.	dox File R	Reference .		 	 	 			 					 16
8.8	workflo	w.dox File	Reference		 	 	 			 					 16
la des															47
Index															17

Porthos

This project tries to create an embedded linux robot system.

A high-level description of the system is given in the system_description. Some requirements have been created, but these are still very much open to discussion.

The compiled documentation can be read at http://spoorcc.github.io/porthos/ And the PDF at http://spoorcc.github.io/porthos/Porthos.pdf

Compiling

mkdir bld
cd bld
cmake ..
make

Generating documentation

cd bld make doc 2 **Porthos**

Licensing

This page describes the licencing for the Porthos system.

2.1 Licensing of documentation

Todo determine licensing

2.2 Licensing of source code

Todo determine licensing

4 Licensing

System Description

Describes the robot system on high-level

6 System Description

Testing

This page describes the testing procedures for the Porthos project.

4.1 Unittests

This project uses check for testing the C-code In order to run the tests do the following

```
cd bld
cmake ..
make
make test
```

To have more output for analyzing failing tests use following command instead of make test:

```
ctest --verbose
```

Testing 8

Workflow

This page describes the workflow for the Porthos project.

5.1 Workflow

This project works following the git-flow branching model. Each feature is developed on a feature branch, branched of of develop. Check out http://nvie.com/posts/a-successful-git-branching-model/ for more info.

The below workflow is based on http://qq.is/tutorial/2011/10/23/git-flow-on-github. \leftarrow html

5.2 Setting up

First clone the repository

git clone https://github.com/spoorcc/porthos.git

Go into the repo

cd porthos

Setup the origin

git remote add upstream git@github.com:spoorcc@porthos

Setup git flow (first install git flow if you haven't got it)

git flow init

And accept all the defaults

5.3 Starting on your feature

Create a new branch for your awesome feature

```
git flow feature start <my_great_feature>
```

Push the branch remote.

10 Workflow

```
git flow feature publish <my_great_feature>
```

Commit your changes reguraly locally with descriptive messages.

Also push the changes back up to GitHub.

```
git push origin HEAD
```

5.4 Finish work

Create a pull request in the GitHub interface. In the pull request add usefull info. Click the send pull request to confirm you think you're done.

When your awesome feature is reviewed, sometimes additional changes are needed. Make them locally, commit and push them up to your branch.

Make sure your on your feature branch:

```
git checkout feature/<my_awesome_feature>
```

Do your development, commit and push the changes again. (see Starting on your feature).

5.5 Cleanup

When all your changes are agreed upon and merged by the project, your feature branch will be deleted. Locally you can finish your feature as well.

```
git flow feature finish
```

Todo List

Page Licensing

determine licensing determine licensing

12 **Todo List**

File Index

7	4	F-11	- 1	1:4
/	1	- 1-11	ו באו	I ICT

Here is a	list of al	I files with	brief	descriptions
i ici c is a	iist oi ai	I IIICS WILLI	DITICI	ucaci iptiona

motion.c	 15
motion.h	 15
test_libmotion.c	 16

14 File Index

File Documentation

8.1 licensing.dox File Reference

8.2 motion.c File Reference

```
#include <stdio.h>
```

Functions

int motion_init ()
 Initializes the motion library.

8.2.1 Function Documentation

```
8.2.1.1 motion_init()
```

```
int motion_init ( )
```

Initializes the motion library.

The motion library is initialized and ready to use.

8.3 motion.h File Reference

```
#include <stdio.h>
```

Functions

int motion_init ()
 Initializes the motion library.

8.3.1 Function Documentation

16 File Documentation

```
8.3.1.1 motion_init()
```

```
int motion_init ( )
```

Initializes the motion library.

The motion library is initialized and ready to use.

- 8.4 README.md File Reference
- 8.5 system_description.dox File Reference
- 8.6 test_libmotion.c File Reference

```
#include <check.h>
#include <stdio.h>
#include "motion.h"
```

Functions

- void test_init (void)
- Suite * motion (void)
- int main (int argc, char *argv[])

8.6.1 Function Documentation

```
8.6.1.1 main()
```

```
int main (
          int argc,
          char * argv[] )
```

8.6.1.2 motion()

```
Suite* motion ( \label{eq:void} \text{void} \quad )
```

8.6.1.3 test_init()

```
void test_init (
     void )
```

- 8.7 testing.dox File Reference
- 8.8 workflow.dox File Reference

Index

```
licensing.dox, 15
main
    test_libmotion.c, 16
motion
    test_libmotion.c, 16
motion.c, 15
    motion_init, 15
motion.h, 15
    motion_init, 15
motion_init
    motion.c, 15
    motion.h, 15
README.md, 16
system_description.dox, 16
test_init
    test_libmotion.c, 16
test_libmotion.c, 16
    main, 16
    motion, 16
    test_init, 16
testing.dox, 16
workflow.dox, 16
```