

# FRC 2018 Software Documentation

Team 5572: The ROSBOTS



# Contents

<b>1</b>	<b>Namespace Index</b>	<b>1</b>
1.1	Namespace List . . . . .	1
<b>2</b>	<b>Class Index</b>	<b>3</b>
2.1	Class List . . . . .	3
<b>3</b>	<b>File Index</b>	<b>5</b>
3.1	File List . . . . .	5
<b>4</b>	<b>Namespace Documentation</b>	<b>7</b>
4.1	field Namespace Reference . . . . .	7
4.2	field::side Namespace Reference . . . . .	7
4.2.1	Function Documentation . . . . .	7
4.2.1.1	scale() . . . . .	7
4.2.1.2	setup() . . . . .	7
4.2.1.3	switch_far() . . . . .	7
4.2.1.4	switch_near() . . . . .	7
4.2.2	Variable Documentation . . . . .	7
4.2.2.1	left . . . . .	7
4.2.2.2	right . . . . .	7

<b>5</b>	<b>Class Documentation</b>	<b>9</b>
5.1	Curve Struct Reference	9
5.1.1	Detailed Description	9
5.1.2	Member Data Documentation	9
5.1.2.1	heading	9
5.1.2.2	x	9
5.1.2.3	y	10
5.2	DoublePair Struct Reference	10
5.2.1	Detailed Description	10
5.2.2	Member Data Documentation	10
5.2.2.1	u	10
5.2.2.2	v	10
<b>6</b>	<b>File Documentation</b>	<b>11</b>
6.1	src/drivetrain.h File Reference	11
6.1.1	Macro Definition Documentation	12
6.1.1.1	HALF_PI	12
6.1.2	Function Documentation	12
6.1.2.1	curveamount(double L1, double L2, double W)	12
6.1.2.2	operator+(Curve a, Curve b)	12
6.2	src/test.cpp File Reference	13
6.2.1	Macro Definition Documentation	13
6.2.1.1	ROOT_PASS	13
6.2.1.2	ROOT_REQUIRED	13
6.2.2	Function Documentation	13
6.2.2.1	main()	13
6.3	src/utls/kernel_interface.h File Reference	13
6.4	src/utls/linux.cpp File Reference	14
6.4.1	Macro Definition Documentation	14
6.4.1.1	READ_END	14
6.4.1.2	ROOT_NAME	14
6.4.1.3	WRITE_END	14
6.4.2	Function Documentation	14
6.4.2.1	root_mode(char *argv[], const char *password)	14
6.4.2.2	signal_register(int a, void(*v)(int))	14
6.5	src/utls/matchdata.h File Reference	15
	<b>Index</b>	<b>17</b>

# Chapter 1

## Namespace Index

### 1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

<a href="#">field</a>	.....	<a href="#">7</a>
<a href="#">field::side</a>	.....	<a href="#">7</a>



## Chapter 2

# Class Index

### 2.1 Class List

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

<a href="#">Curve</a>	Describes the position and direction of a robot after a curve amount . . . . .	<a href="#">9</a>
<a href="#">DoublePair</a>	Stores generic 2-value real number objects . . . . .	<a href="#">10</a>





## Chapter 3

# File Index

### 3.1 File List

Here is a list of all files with brief descriptions:

src/ <a href="#">drivetrain.h</a>	11
src/ <a href="#">test.cpp</a>	13
src/utls/ <a href="#">kernel_interface.h</a>	13
src/utls/ <a href="#">linux.cpp</a>	14
src/utls/ <a href="#">matchdata.h</a>	15



## Chapter 4

# Namespace Documentation

### 4.1 field Namespace Reference

#### Namespaces

- [side](#)

### 4.2 field::side Namespace Reference

#### Functions

- void [setup](#) ()
- bool [switch\\_near](#) ()
- bool [switch\\_far](#) ()
- bool [scale](#) ()

#### Variables

- const bool [left](#) = false
- const bool [right](#) = true

#### 4.2.1 Function Documentation

4.2.1.1 `bool field::side::scale ( )` `[inline]`

4.2.1.2 `void field::side::setup ( )` `[inline]`

4.2.1.3 `bool field::side::switch_far ( )` `[inline]`

4.2.1.4 `bool field::side::switch_near ( )` `[inline]`

#### 4.2.2 Variable Documentation

4.2.2.1 `const bool field::side::left = false`

4.2.2.2 `const bool field::side::right = true`



# Chapter 5

## Class Documentation

### 5.1 Curve Struct Reference

Describes the position and direction of a robot after a curve amount.

```
#include <drivetrain.h>
```

#### Public Attributes

- double `x`  
*Horizontal Position.*
- double `y`  
*Vertical Position.*
- double `heading`  
*Direction of the robot in radians.*

#### 5.1.1 Detailed Description

Describes the position and direction of a robot after a curve amount.

#### 5.1.2 Member Data Documentation

##### 5.1.2.1 double `Curve::heading`

Direction of the robot in radians.

##### 5.1.2.2 double `Curve::x`

Horizontal Position.

### 5.1.2.3 double Curve::y

Vertical Position.

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

- [src/drivetrain.h](#)

## 5.2 DoublePair Struct Reference

Stores generic 2-value real number objects.

```
#include <drivetrain.h>
```

### Public Attributes

- double [u](#)  
*First Value.*
- double [v](#)  
*Second Value.*

### 5.2.1 Detailed Description

Stores generic 2-value real number objects.

Examples of usage are 2d coordinates and differential drive outputs.

### 5.2.2 Member Data Documentation

#### 5.2.2.1 double DoublePair::u

First Value.

#### 5.2.2.2 double DoublePair::v

Second Value.

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

- [src/drivetrain.h](#)

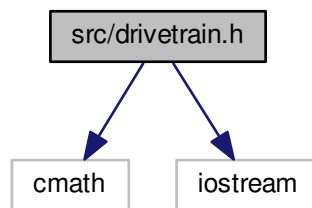
## Chapter 6

# File Documentation

### 6.1 src/drivetrain.h File Reference

```
#include <cmath>
#include <iostream>
```

Include dependency graph for drivetrain.h:



#### Classes

- struct [Curve](#)  
*Describes the position and direction of a robot after a curve amount.*
- struct [DoublePair](#)  
*Stores generic 2-value real number objects.*

#### Macros

- `#define` [HALF\\_PI](#) 1.570796327

#### Functions

- [Curve operator+](#) ([Curve](#) a, [Curve](#) b)  
*Adds two curves.*
- [Curve curveamount](#) (double L1, double L2, double W)  
*Turn two encoder readings (in coordinate units) into coordinates.*

### 6.1.1 Macro Definition Documentation

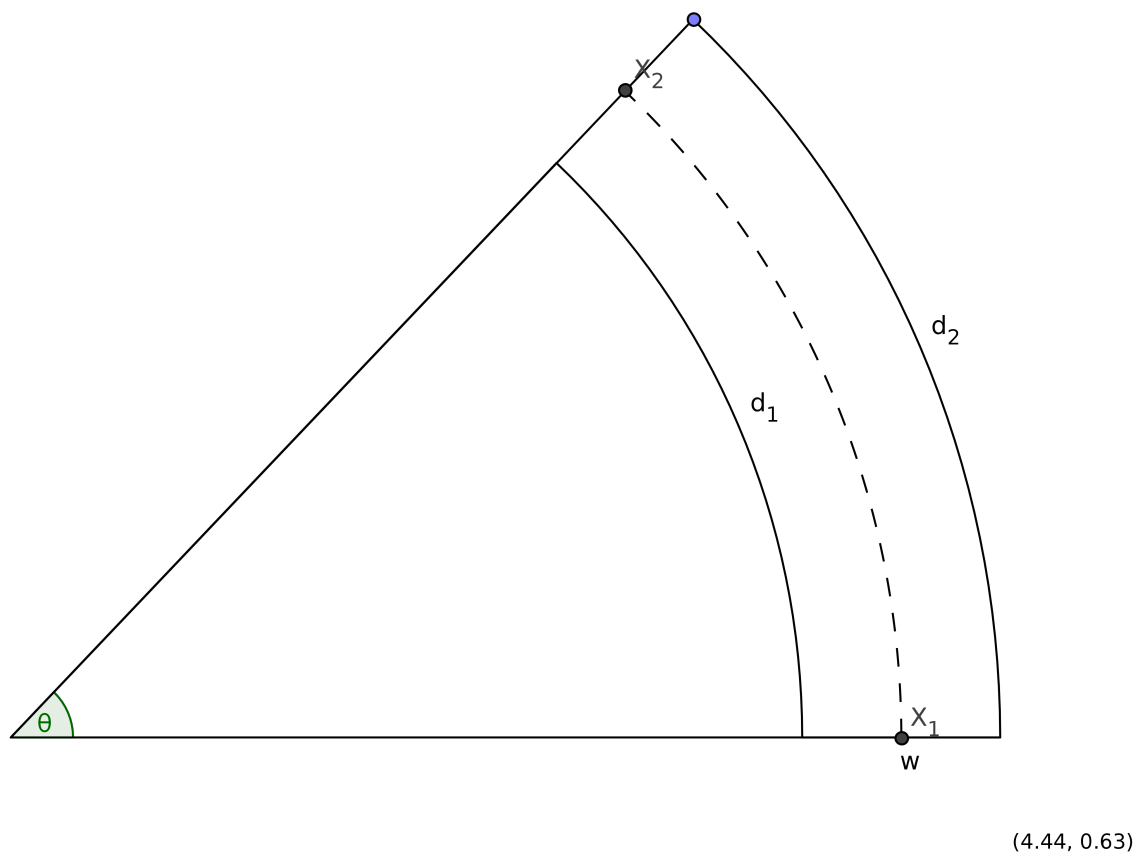
6.1.1.1 `#define HALF_PI 1.570796327`

### 6.1.2 Function Documentation

6.1.2.1 **Curve** `curveamount ( double L1, double L2, double W )` `[inline]`

Turn two encoder readings (in coordinate units) into coordinates.

(0.77, 3.44)



$$r = \frac{y}{\sin(\theta)} \quad (6.1)$$

$$x = \frac{y}{\sin(\theta)} \quad (6.2)$$

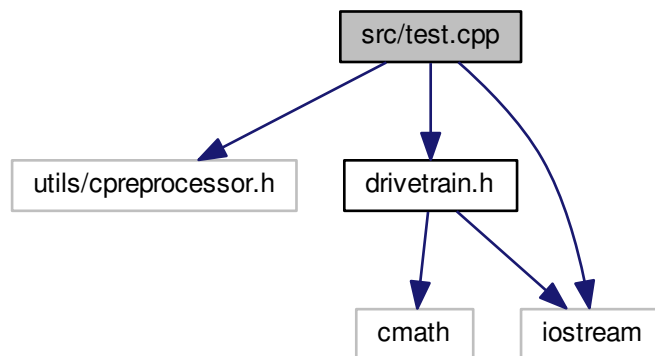
6.1.2.2 **Curve** `operator+ ( Curve a, Curve b )` `[inline]`

Adds two curves.



## 6.2 src/test.cpp File Reference

```
#include "utils/cpreprocessor.h"  
#include "utils/kernel_interface.h"  
#include <iostream>  
Include dependency graph for test.cpp:
```



### Macros

- `#define` `ROOT_REQUIRED`
- `#define` `ROOT_PASS` "socrates"

### Functions

- `int` `main` ()

#### 6.2.1 Macro Definition Documentation

6.2.1.1 `#define` `ROOT_PASS` "socrates"

6.2.1.2 `#define` `ROOT_REQUIRED`

#### 6.2.2 Function Documentation

6.2.2.1 `int` `main` ( )

## 6.3 src/utils/kernel\_interface.h File Reference

This graph shows which files directly or indirectly include this file:

## 6.4 src/utls/linux.cpp File Reference

```
#include "kernel_interface.h"
#include <unistd.h>
#include <iostream>
#include <string.h>
#include <pwd.h>
#include <sys/wait.h>
#include <stdio.h>
#include <signal.h>
```

Include dependency graph for linux.cpp:

### Macros

- #define [ROOT\\_NAME](#) "root"
- #define [WRITE\\_END](#) 1
- #define [READ\\_END](#) 0

### Functions

- void [signal\\_register](#) (int a, void(\*v)(int))  
*Creates a signal handler.*
- void [root\\_mode](#) (char \*argv[], const char \*password)  
*Resets program to be in sudo mode.*

### 6.4.1 Macro Definition Documentation

#### 6.4.1.1 #define READ\_END 0

#### 6.4.1.2 #define ROOT\_NAME "root"

#### 6.4.1.3 #define WRITE\_END 1

### 6.4.2 Function Documentation

#### 6.4.2.1 void root\_mode ( char \* argv[], const char \* password )

Resets program to be in sudo mode.

If the program is already root, it will instead just return

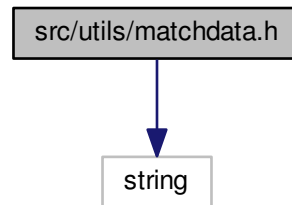
#### 6.4.2.2 void signal\_register ( int a, void(\*)(int) v )

Creates a signal handler.

## 6.5 src/utils/matchdata.h File Reference

```
#include <string>
```

Include dependency graph for matchdata.h:



### Namespaces

- `field`
- `field::side`

### Functions

- void `field::side::setup ()`
- bool `field::side::switch_near ()`
- bool `field::side::switch_far ()`
- bool `field::side::scale ()`

### Variables

- const bool `field::side::left` = false
- const bool `field::side::right` = true



# Index

- Curve, 9
  - heading, 9
  - x, 9
  - y, 9
- curveamount
  - drivetrain.h, 12
- DoublePair, 10
  - u, 10
  - v, 10
- drivetrain.h
  - curveamount, 12
  - HALF\_PI, 12
  - operator+, 12
- field, 7
- field::side, 7
  - left, 7
  - right, 7
  - scale, 7
  - setup, 7
  - switch\_far, 7
  - switch\_near, 7
- HALF\_PI
  - drivetrain.h, 12
- heading
  - Curve, 9
- left
  - field::side, 7
- linux.cpp
  - READ\_END, 14
  - ROOT\_NAME, 14
  - root\_mode, 14
  - signal\_register, 14
  - WRITE\_END, 14
- main
  - test.cpp, 13
- operator+
  - drivetrain.h, 12
- READ\_END
  - linux.cpp, 14
- ROOT\_NAME
  - linux.cpp, 14
- ROOT\_PASS
  - test.cpp, 13
- ROOT\_REQUIRED
  - test.cpp, 13
- right
  - field::side, 7
- root\_mode
  - linux.cpp, 14
- scale
  - field::side, 7
- setup
  - field::side, 7
- signal\_register
  - linux.cpp, 14
- src/drivetrain.h, 11
- src/test.cpp, 13
- src/Utils/kernel\_interface.h, 13
- src/Utils/linux.cpp, 14
- src/Utils/matchdata.h, 15
- switch\_far
  - field::side, 7
- switch\_near
  - field::side, 7
- test.cpp
  - main, 13
  - ROOT\_PASS, 13
  - ROOT\_REQUIRED, 13
- u
  - DoublePair, 10
- v
  - DoublePair, 10
- WRITE\_END
  - linux.cpp, 14
- x
  - Curve, 9
- y
  - Curve, 9